A Methodical Framework for Engineering Co-evolution for Simulating Socio-economic Game Playing Agents

Arjun Chandra

School of Computer Science

University of Birmingham

A thesis submitted for the degree of

Doctor of Philosophy

October 2011

Abstract

Agent based computational economics (ACE), as a research field, has been using co-evolutionary algorithms for modelling the socio-economic learning and adaptation process of players within games that model socio-economic interactions. In addition, it has also been using these algorithms for optimising towards the game equilibria via socio-economic learning. However, the field has been diverging from evolutionary computation, specifically co-evolutionary algorithm design research. It is common practice in ACE to explain the process and outcomes of such co-evolutionary simulations in socio-economic terms. However, co-evolutionary algorithms are known to have unexpected dynamics that lead to unexpected outcomes. This has often lead to mis-interpretations of the process and outcomes in socio-economic terms, a case in point being the lack of a methodical use of the term *bounded rationality*. This mis-interpretation can be attributed to the lack of a proper consideration of the solution concept being implemented by the co-evolutionary algorithm used for the simulation.

We propose a holistic methodical framework for analysing and designing co-evolut- ionary simulations, such that mis-interpretations of socio-economic phenomena be methodically avoided, disabling the algorithm from being mis-interpreted in socio-economic terms, aimed at benefiting ACE as a research field. More specifically, we consider the methodical treatment of co-evolutionary algorithms, as enabled by the framework, such that mis-interpretations of bounded rationality be avoided when these algorithms are used to optimise towards equilibrium solutions in bargaining games. The framework can be broken down into two parts:

- Analysing and refining co-evolution for ACE, using the notion behind co-evolut- ionary solution concepts from co-evolutionary algorithm design research: Challenging the value of the implicit assumption of bounded rationality within co-evolutionary simulations, which leads to it being mis-interpreted, we show that

convergence to the equilibrium solutions can be achieved with boundedly rational agents by working on the elements of the implemented co-evolutionary solution concept, as opposed to previous studies where bounded rationality was seen as the cause for deviations from equilibrium. Analysis and refinements guided by the presence of top-down equilibrium solutions, allow for a *top-down avoidance of mis-interpretations* of bounded rationality within simulations.

- Analysing and refining co-evolution for ACE, using the notion behind reconciliation variables proposed in the thesis: Reasonably associating mis-interpreted socio-economic phenomena of interest with the elements of the implemented co-evolutionary solution concept, parametrising and quantifying the elements, we obtain our reconciliation variables. Systematically analysing the simulation for its relationship with the reconciliation variables or for its closeness to desired behaviour, using this parametrisation, is the suggested idea. Bounded rationality is taken as a reconciliation variable, reasonably associated with agent strategies, parametrised and quantified, and analysis of simulations with respect to this variable carried out. Analysis and refinements based on such an explicit expression of bounded rationality, as opposed to the erstwhile implicit assumption, allow for a *bottom-up avoidance of mis-interpretations* of bounded rationality within simulations.

We thus remove the causes that lead to bounded rationality being mis-interpreted altogether using this framework. We see this framework as one next step in ACE socio-economic learning simulation research, which must not be overlooked.

For my parents.

Acknowledgements

I would like to acknowledge some people without whom, I believe with full certainty, that this thesis would not take shape in the form it is. Thanks for putting up with me.

- My supervisor, Xin Yao, for giving me the opportunity, helping promptly whenever I asked, and never asking for much in return. My thesis group, Jonathan Rowe, Peter Tino and Colin Rowat, who I believe have been very patient with me. Peter Hancox and Steve Vickers for listening to me. The Market Based Control Project for an experience of a lifetime.
- Graham Kendall, Peter Tino again and Behzad Bordbar, for an enjoyable defence experience.
- Pietro Oliveto for single handedly putting me back on track with Science in general and the PhD research in particular, for being a very good friend, and for being my *Brummie family*.
- Kai Kurihara and Melissa Short for being very good friends and my *Mancunian family*. Thanks for listening to me, and just being there.
- Gavin Brown for helping me make the right choice when things were very uncertain and help me carry on with research.
- Stefan Haflidason and Richard Allmendinger for being *very big inspirations*, very good friends, and for extremely useful discussions. If anyone wants to feel high on *energy for getting things done*, I can send them to these guys. Further thanks to Richard (volunteering - talking about energy) for proof reading.
- Loretta Mancini, Josslin Noirel, Kyriakos Anastasakis, Philipp Rohlfshagen, Grace Rohlfshagen, Stian Soiland-Reyes, Claudia Soiland-Reyes, Luke Thomas, Tamsin Thomas, Sule Guney, Lydia Nyilasi and Kateryna Vyshnyak for being very good friends, and for their support in various forms at various moments towards the completion of this thesis.

- Peter Lewis, Edward Robinson, Vivek Nallur and Yuliya Tarabalka for very useful discussions, specially in the final months, and being very good friends.
- Peter McBurney, Bruce Edmonds and the Manchester Complexity Group for very useful discussions.
- The Machine Learning and Optimisation group at University of Manchester for keeping me academically aloft.
- The excellent work environment at the University of Oslo, and the excellent new colleagues I am just about starting to work with, the combination of which helped a lot during the thesis modification period.
- Iain Dixon, Marcin Osinski, Karolina Bojakowska, Marek Dudzinski, Sigtriggur Sigtriggson and the University of Manchester Gliding Society for being very good friends and keeping me sane with the much needed fuel from the outdoors, that I could even make a career out of.
- Tina Wenzel for just being there, and indeed during the home stretch.
- And, finally, my family, and specially my Parents, for their patience and support in all these years I have been away from home. My Father, for handling my uncertain situation (which I know all too well how it affects him) with a lot of patience, and my Mother, for being the strongest person I have known. This thesis is for you both.

In all, Birmingham gave me an opportunity and lessons for life, Manchester helped me realise and execute what I learnt, and the above mentioned supported me through it all.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

> *Co-evolutionary algorithms often exhibit unexpected dynamics that lead to unexpected outcomes. These algorithms are used for modelling the socio-economic learning process within the field of agent based computational economics (ACE), thus simulating the learning and adaptation process of players or agents within socio-economic games. Interpreting the co-evolutionary process and outcomes in socio-economic terms, when using these algorithms off-the-shelf as is often the case, results in the algorithm being mis-interpreted in socio-economic terms. For example, the term bounded rationality is generally assumed to be an implicit property of these algorithms and the results from the algorithms mis-interpreted based on it. How might we scrutinise co-evolutionary algorithms for carrying out socio-economic simulations so that such mis-interpretations be methodically avoided?*

This is essentially what this thesis is about. There is a need for understanding and scrutinising co-evolutionary algorithms in a methodical manner such that they can be better understood as models of socio-economic learning, with a view that they may not be mis-interpreted in socio-economic terms.

Agent based computational economics (ACE), as a research field, has ardently been using co-evolutionary algorithms for socio-economic simulations for various

purposes. Some of these are, modelling socio-economic learning and adaptation [1, 5, 39, 59], modelling market mechanisms [94], understanding the emergence of behavioural norms [12, 159], understanding the emergence of market structure [144], amongst others. The choice of the algorithm for the above mentioned purposes is however ad-hoc [39] or off-the-shelf [145]. There is empirical evidence in favour of the use of co-evolutionary algorithms for simulating socio-economic learning, i.e sometimes they seem to satisfy expectation [39], for example, if we want them to evolve towards the equilibrium solutions or model human behaviour within socio-economic games, they seem to suggest that sometimes they can. Quoting Dawid [39] commenting on some empirical work with genetic algorithms used to simulate socio-economic learning, *"The fact that all these comparisons yield positive results may be an indicator that some of the features of genetic algorithms are indeed a good representation of effects appearing in learning populations"*. However, there is a gap that still remains. It has been firmly expressed that the *"unconsidered adoption"* [145] of co-evolutionary algorithms for socio-economic simulations is problematic [1, 39, 145]. When simulating socio-economic learning, it is generally accepted by the ACE research practitioners to give a socio-economic interpretation to the algorithm considered, which, in our case, would be an off-the-shelf co-evolutionary algorithm. For example, socio-economic terms are used to describe the intricacies of the algorithm, a running example throughout the thesis being the use of the term bounded rationality to describe the behaviour of agents as they learn and act by trial and error. As such, the algorithm is treated as explaining or modelling socio-economic phenomena directly or with little consideration for a rigorous match with the socio-economic phenomena it is supposed to model to be established first. This can lead to mis-interpretations of the algorithm in the socio-economic context. It is easy to understand why there may be mis-interpretations.

As far as a methodical choice of an algorithm goes, it is common within the field of ACE to consider one of the two class of algorithms (also known as levels of learning) for the purpose of simulations, these classes being *individual learning* and *social learning*. These two classes stem from the initial work carried out in the field of evolutionary computation, more specifically evolutionary learning, whereby two approaches (or schools of thought) known as the *Michigan* [74] and

Pittsburg [42] approach to evolutionary learning were made prominent. These approaches differ in the way a solution to the problem is constituted from an evolving population of solutions. In the Michigan approach, the population as a whole constitutes a solution (individual being partial solutions), whereas in the Pittsburg approach, each individual in the evolving population constitutes a solution to the problem. Considering one agent (or the strategy used by the agent to play a game) as being this solution discussed above, these two approaches directly map to the individual learning and social learning classes commonly considered to make a choice of an algorithm within ACE. Choosing an algorithm residing in one of these two classes, even because of a class being more popular amongst practitioners [152], rather than a due to a rigorous study backing the usage, is not uncommon. Thus, there is a lack of a methodology for making a choice of an algorithm, rendering this choice to generally be off-the-shelf. Given the off-the-shelf nature of the application of these algorithms, it is hard to say with any confidence that a particular interpretation is reliable. Moreover, it has further been shown that there is a need for considering socio-economic and algorithmic parameters separately, for certain algorithmic parameters may not have a direct socio-economic interpretation (for example, changing the population size can change the dynamic and outcomes from the simulation [2], specifically when using the social learning class of algorithms, as described above, for simulations). The simulations should thus be robust to changes in values of such parameters [1].

In addition, co-evolutionary algorithm design, as a research field, has grown largely independently of ACE. The link between the two fields, barring a few exceptions [1, 8, 39, 96], has only been the application of these algorithms in ACE, and using socio-economic problems considered in ACE as test beds to study and understand co-evolutionary algorithms, without informing ACE[1] (in particular,

[1]Much of the work presented in this thesis links the notion behind co-evolutionary solution concepts (see Glossary for a definition) from co-evolutionary algorithm design research, considered by Ficici in his PhD thesis [53], and the application of an off-the-shelf co-evolutionary algorithm within ACE for the case of bargaining games, which was considered by Gerding in his PhD thesis [58]. Both these theses were awarded in the same year, i.e. 2004, and do not cite each other. They both however cite one of the early works considering the link between co-evolution and ACE [39] (from 1996) ephemerally, which in fact argues against the 'ad-hoc' usage of co-evolutionary algorithms within ACE. As such, there has clearly been a divergence

the sub field of modelling socio-economic learning and adaptation). The field of co-evolutionary algorithm design has discovered many issues that these algorithms have, from the point of view of them doing what is expected of them. For example, there may be intransitivities in the search space such that the algorithm may end up cycling within the space [106], amongst other issues (further issues with these algorithms are discussed in Chapter 2). These issues were mostly discovered due the co-evolutionary algorithm in question behaving unexpectedly. The unexpected dynamics and results from co-evolutionary algorithms necessitate deeper analysis of these algorithms, as opposed to prematurely using socio-economic terms to explain the workings of the algorithms. Such explanations can easily lead to mis-interpreting the algorithms in socio-economic terms, as these issues may seep into simulations with an ad-hoc approach to the usage of these algorithms. All these issues within co-evolutionary algorithms have now been described as branching out of one essential issue with co-evolution in [53]. This issue is the lack of rigour in the definition of a solution concept that co-evolutionary algorithms are meant to implement. According to [53], every search algorithm implements a solution concept. In other words, every search/learning problem has a solution concept associated with it, and the solution concept is realised by implementing a search algorithm. It would then seem that it is the solution concept implemented that must be scrutinised before interpreting the algorithm in socio-economic terms, or there is a danger of mis-interpreting the algorithm, and indeed, mis-interpreting or mis-representing socio-economic terms within the simulation, i.e. mis-interpreting the role of the socio-economic phenomena that are denoted by the terms within the simulation.

A solution concept is a notion borrowed from game theory. It specifies whether a strategy played by an agent adheres to a set of standards that make the strategy preferable to be used in a problem over other strategies, and thus, retained as a solution to the problem. It separates solution strategies from those that are not preferred or desired as solutions, almost like a filter through which the whole search space of strategies is passed, retaining only the desired strategy or strategies. We do not have the luxury of explicitly scrutinising each strategy in the search space, which leads to sampling strategies (via exploration of the strategy

in the two fields.

space), evaluating them, and following a gradient in the search space towards the desired solution strategies. The solution concept can thus be specified as using the gradient to separate potentially preferable strategies from those that are not. Co-evolution, which helps specify the solution concept and thus helps solve the search problem of finding the desired solutions, makes specifying this gradient harder however, because the evaluation of strategies is relative to the strategies sampled, and not against a static/absolute function that needs optimising. The pressure towards filtering out the desired strategies is built on by channeling the strategies along a certain path in the search space. This path in the search space is however laid out based on an incomplete and changing view of the space, the current (at any point during the course of the search process) strategies themselves representing the incomplete view of the search space. This is very important in co-evolution because the incomplete view of the search space is also *reactive*, in the sense that it adapts, to the strategies representing the current state of the algorithm, i.e it is *important to find which strategies to interact with so that these strategies (used for the interaction, and in turn, as opponents for evaluation) react or adapt favourably/usefully towards laying out the path.* A secondary search problem (requiring search effort) thus results, that of searching who the strategies must interact with. The opponent strategies that describe the incomplete yet useful view of the search space to adapt with or evaluate against, need attention in their discovery and use. Most of the problems in co-evolution stem from a lack of rigour in designing the elements of co-evolution that make up the algorithm, which then implements the solution concept it is aimed at, such that the requisite search effort is indeed expended.

Our view in this thesis is that there is a dissonance between *interpreting co-evolution as a socio-economic simulator* and *viewing co-evolution as a search process.* When taking co-evolutionary algorithms as simulators, *socio-economic phenomena* like *bounded rationality* (which happens to be a widely accepted phenomenon for describing human decision making in economics since [138]), have been assumed as a hindrance to these simulations achieving the desired goals, *without much scrutiny.* According to [21], these simulations do involve soft factors that are difficult to quantify, for instance bounded rationality, yet they are seen as the *"only game in town"* [21] in order to model the situation at hand,

which in our case would be a socio-economic one. However, these algorithms do provide for the flexibility to tune the complexity of agents, including their degree of rationality [21], and understand the affect of this complexity on the emergent process and outcomes, apart from the algorithms themselves having the flexibility to be tuned per se. Still, bounded rationality is generally assumed within co-evolutionary algorithmic simulations of socio-economic situations. A recent comprehensive survey of evolutionary methods, when used as socio-economic simulators, corroborates this practice [131]. Essentially, bounded rationality is assumed to be implicit in these simulations, i.e. the specific choices made about the components of the algorithm are assumed to define it [131]. Such lack of scrutiny leads to mis-interpretations. On the other hand, when taking these algorithms as a search process, the multiple issues that are now seen as due to the lack of rigour in the proper definition of the co-evolutionary solution concept, are seen as a hindrance to co-evolution achieving the desired goals. It is clear that given co-evolutionary algorithms already have issues, they will not go away if they are taken off-the-shelf, with little scrutiny, and used for socio-economic simulations. Hence the dissonance.

The former (interpreting co-evolution as a socio-economic simulator) has not considered refining the co-evolutionary process in a methodical manner when it comes to socio-economic interpretations of co-evolutionary algorithms. Socio-economic phenomena may not have an obvious algorithmic interpretation and thus the algorithm may have to be refined. Note that this is in contrast to [1], where algorithm parameters may not have a socio-economic interpretation and so one needs to be careful about the robustness of the algorithms to a change in values of these parameters. Also, although simulations can be calibrated [96] to real world data, we may not have enough data for such calibrations in reality, thus bringing in other issues of research to be dealt with in case one goes this route. In any case, refining the process at a fundamental algorithmic level in a more methodical manner has not been much explored within ACE. Refinements have been looked at from purely the algorithmic viewpoint in [8], but the idea there was to compare two off-the-shelf algorithms and understand how one can inform and emulate the other, instead of dealing with assumed socio-economic phenomena that can lead to socio-economic mis-interpretations of simulations.

The latter (viewing co-evolution as a search process), does promote the idea of refinements to the search such that the envisaged goals are indeed achievable [53]. One way this can be done is by considering the notion behind co-evolutionary solution concepts and tangibly working on the elements that the algorithm is composed of in a systematic manner, as we will show in this thesis in Chapter 3. This enables us to understand the reasons for discrepancies between the desired and actual outcomes from the co-evolutionary process, and thus understand what the problems are, followed by tackling them. One cannot tackle a problem if one does not know what the problem is in the first place. *The notion behind co-evolutionary solution concepts provides a way of thinking about refinements to the algorithmic workings and systematic disciplining of co-evolution.*

Game theory provides a powerful framework to understand interactions between entities. These entities could be players with complete knowledge of their environment, players with incomplete knowledge of the environment, and more pertinently, computational agents interacting with each other under various assumptions about their deliberative abilities. Theoretical results from game theory, e.g. a single selected Nash Equilibrium or the Subgame Perfect Equilibrium (SPE) [133] as it is called in a sequential game (which we will consider in detail in Chapter 2), give us a useful idealised result in terms of what outcomes one may expect with players interacting under theoretical assumptions of perfect rationality and complete knowledge. Although these theoretical assumption are far too strict and do not in any way model the reality of the particular interaction or game under question, they are indeed required for a closed form or top-down solution to the game. The value of these theoretical outcomes is arguable, at least in economics and psychology research. However, for the purpose of understanding and disciplining bottom-up approaches like co-evolution, they are indeed a boon. These solutions give us a solution concept to aim for, thus enabling validating the simulations. Be it simulating real world socio-economic learning and outcomes thereof, or simulating equilibrium selection via socio-economic learning, using co-evolutionary algorithms in a more methodical manner applies to both. The advantage with equilibrium selection is for there being a solution concept at hand, whereas with real world problems, one has to deal with experiments with (or at least experimental data from socio-economic games with) humans,

and that is beyond the scope of this thesis.

As a *case study*, we focus on *bounded rationality* as the socio-economic phenomenon of interest causing a dissonance between interpreting co-evolution as a socio-economic simulator and viewing co-evolution as a search process, thus being mis-interpreted in simulations. This is done to elaborate on and *establish the link* between co-evolutionary algorithm design research and socio-economic learning, such that the algorithms be tangibly refined in a more methodical manner for their application in the latter. We do this in the context of *bargaining games*. These games have been considered in ACE in order to simulate convergence to equilibrium solutions (more specifically the subgame perfect equilibrium) via socio-economic learning and adaptation using co-evolutionary algorithms [58]. Bounded rationality was seen as the cause for the co-evolutionary algorithm deviating from the desired goal of converging to the subgame perfect equilibrium solution with little scrutiny, i.e. it was *assumed* (both as implicit in the simulation, as generally done in ACE, and as the cause for deviations), hence attracting interest from the point of view of this thesis. Moreover, this game has had many applications in real world situations, e.g. bargaining problems concerning international trade, industrial organisation, political economy etc. [135]. This gives further motivation for us to consider it.

1.2 Objectives

We want to lay down a methodical framework for analysing and refining (also referred to as engineering in the thesis) co-evolutionary algorithms, when used as models of socio-economic learning within ACE simulations. The main characteristic of the framework is for it to be used for analysing and refining these simulations, such that there is no room left for mis-interpretations of socio-economic phenomena, which are often used to explain the co-evolutionary process or outcomes, within simulations. As a result, the co-evolutionary process and outcomes are not allowed to be mis-interpreted in socio-economic terms which were being mis-interpreted within simulations in the first place.

We want to utilise this framework to remove the causes that may lead to mis-interpreting bounded rationality (the socio-economic phenomenon of interest to

us) within co-evolutionary simulations altogether. We want to do this in the context of using co-evolutionary algorithms as modelling socio-economic learning to optimise towards the equilibrium solutions in bargaining games. The framework is composed of two integral parts, which together explain the framework, laying out and applying them forming the main objective of this research. These two can, in the context of applying the framework for the case of the socio-economic phenomenon of interest of bounded rationality, be stated as follows:

- We want to show how the implicit assumption of bounded rationality within ACE simulations can be challenged for its inadequacy using the notion behind co-evolutionary solution concepts from co-evolutionary algorithm design research, such that it becomes apparent as to what it does not mean in these simulations, suggesting that it was indeed being mis-interpreted, and that the co-evolutionary algorithm was being mis-interpreted in socio-economic terms. We want to show that bounded rationality does not explain the co-evolutionary simulations not converging towards the equilibrium solutions, by making the co-evolutionary simulations converge to these solutions in the presence of bounded rationality. Thus, we want to show how to avoid mis-interpretations of socio-economic phenomena within simulations from the top-down (via guidance provided by top-down desired solutions), using the notion behind co-evolutionary solution concepts. This becomes the idea behind the first integral part of the methodical framework.

- We then want to show how bounded rationality may be explicitly expressed within simulations using the notion behind reconciliation variables that we propose in this thesis, leading to understanding the role of bounded rationality within simulations in explicit algorithmic terms. This shows the explicit hold that we can have on this socio-economic phenomenon of interest, thus leaving no room for its mis-interpretation. Thus, we want to show how to avoid mis-interpretations of socio-economic phenomena within simulations from the bottom-up (via an explicit bottom-up expression of the phenomena within simulations), using the notion behind reconciliation variables. This becomes the idea behind the second integral part of the methodical framework.

The above two mis-interpretation avoidance scenarios holistically specify the framework we want to put forth for the ACE community to follow.

1.3 Contributions

Given the motivation and objectives, we now look at the contributions this thesis makes. Laying out of a holistic methodical framework for co-evolutionary algorithm analysis and refinement, such that mis-interpretations of socio-economic phenomena of interest be avoided within co-evolutionary simulations, forms the main theme of the thesis. The framework is composed of two integral parts, which together specify the framework, and are two essential contributions of the thesis. The framework, thus specified and then applied to previous work, is also a contribution. Following is a summary of the two integral parts and their application towards the avoidance of mis-interpretations of bounded rationality within simulations.

Summary Statement of Part 1 of the Framework:

> *Analysing and refining co-evolution for ACE, using the notion behind co-evolutionary solution concepts, empowered by the existence of top-down solutions, from co-evolutionary algorithm design research. Socio-economic phenomena which may have been mis-interpreted within simulations, can be challenged using this notion, and their mis-interpretation revealed. As such, analysis and refinements of co-evolutionary simulations, guided by the presence of top-down solutions, allow for a* ***top-down avoidance of mis-interpretations*** *of socio-economic phenomena of interest within simulations.*

Application of Part 1 of the Framework:

We look at the implemented co-evolutionary solution concept from previous work [59]. This co-evolutionary solution concept took bounded rationality for granted, assuming it to be causing deviations from the expected behaviour of the algorithm. We show that, by working on the elements of the implemented co-evolutionary solution concept systematically, and scrutinising as to what may

have caused the deviations from the expected behaviour algorithmically, we can say the following:

- We now understand, in explicit algorithmic terms, what the *assumption* of bounded rationality actually means in previous work. Specifically, the design choices for the algorithm result in the aforesaid deviations, the relationship between the design choice and deviations having gone unnoticed in previous work, and the choices implicitly assumed to mean bounded rationality.

- We discover that, changing the implemented co-evolutionary solution concept, specifically changing the variation, interaction and evaluation details, results in achieving the originally envisaged goal of converging to the subgame perfect equilibrium solution. We thus achieve equilibrium selection with boundedly rational agents. In fact, practically speaking, we achieve convergence to the equilibrium solution via simulation for the games in which the previous algorithm could not. As such, we can say that the previous work was not implementing the envisaged solution concept. This tells us the importance of considering the notion behind co-evolutionary solution concepts for ACE research. Not considering it, ACE is in danger of going off track, being mislead, and remaining immature as a field when it comes to the *method of adopting* co-evolutionary algorithms as simulating socio-economic learning.

- In effect, since we do achieve convergence to the equilibrium with boundedly rational agents, we have also understood what bounded rationality, as a socio-economic phenomenon, does not mean in the context of the socio-economic simulations using the co-evolutionary algorithm considered in previous work. Note that although this understanding is with respect to the bargaining games and the co-evolutionary algorithm considered in previous work, the method of understanding the simulations through the lenses of co-evolutionary solutions concepts is generally applicable. It also calls into question the general practice within ACE of leaving bounded rationality unscrutinised and assuming it to be implicitly defined by the specification

of the algorithms used for simulations. Thus, challenging socio-economic assumptions or phenomena of interest from the co-evolutionary solution concept perspective is indeed a methodology worth consideration, and further exploration.

Scaling up to more complex problems incorrectly, mis-interpreting, and thus being mislead, is worse than not scaling up but being correct for the problems an algorithm can solve. We have not looked into how our refined algorithm or simulation scales up to more complex games other than bargaining games with multiple rounds (more details about the games to follow in Chapters 2 and 3) in this thesis, however, it is correct for what it can do. We note that there is still a possibility of bounded rationality getting mis-interpreted within our simulation, because, using the first part of the framework we are only able to say the role bounded rationality does not play in the simulation. Since the focus of the thesis is to avoid mis-interpretations of socio-economic phenomena within simulations in a holistic manner, as laid out in the objectives in Section 1.2, we thus want to show how to understand bounded rationality better within such a simulation, instead of using the term to explain away the non-conformance between simulation and reality (for example, assuming bounded rationality causing deviations not only in simple, but more complex games in previous work [59]), without the simulation being correctly or thoroughly understood.

We are interested in socio-economic simulations, and reality (as verified by experimental economics research) does not conform with rational game theoretic equilibria. In reality, bounded rationality is indeed a cause or at least a *reasonable assumption and indeed a widely accepted phenomenon* to be a cause for deviations from rational outcomes. So, *we take bounded rationality as causing this non-conformance seriously now, i.e. we want to know how to achieve various levels of non-conformance*, in order to systematically understand its relationship with co-evolutionary simulations more deeply, and to analyse the simulations for their closeness to desired outcomes with respect to it. In essence, we want to understand how to grow deviating outcomes, given our algorithm, such that it allows us to understand what bounded rationality means within simulations. Note that we do not match simulation and reality in this thesis, but instead understand the affect that bounded rationality has on the co-evolutionary simulations, which

is a contribution to ACE research in itself, considering our methodology, which we summarise next.

We have an algorithm which tells us how to get to the equilibrium in the presence of implicitly present bounded rationality, so by systematically and explicitly quantifying/expressing and denoting bounded rationality as, what we call, a *'reconciliation variable'*, we can understand how it really affects socio-economic simulations if we want to model and simulate the above non-conformance, and more importantly for the purposes of this thesis, rule out the possibility of it being mis-interpreted within simulations. The fact that we call bounded rationality a reconciliation variable is simply because we want to stress the fact that we show how to reconcile deviations or non-conforming outcomes with simulated outcomes, i.e simulate deviations, by expressing bounded rationality as a variable, wherein, varying bounded rationality causes deviations. The next contribution is thus, the notion and use of reconciliation variables that we propose in this thesis, and the second part of the methodical framework which allows us to use the notion. Following is a summary of the second part and its application towards the avoidance of mis-interpretations of bounded rationality.

Summary Statement of Part 2 of the Framework:

> *Analysing and refining co-evolution for ACE, using the notion behind reconciliation variables proposed in the thesis. Reasonably associating mis-interpreted socio-economic phenomena of interest with the elements of the implemented co-evolutionary solution concept, parametrising and quantifying the elements, we obtain our reconciliation variables. Systematically analysing the simulation for its relationship with the reconciliation variables or for its closeness to desired behaviour, using this parametrisation, is the suggested idea. Analysis and refinements based on such an explicit expression of the socio-economic phenomena of interest, allow for a* ***bottom-up avoidance of mis-interpretations*** *of the phenomena within simulations.*

Application of Part 2 of the Framework:

We thus explore bounded rationality as the phenomenon of interest even further. A *methodology* is presented which allows *tangible and systematic handling* of the co-evolutionary algorithm from the point of view of the phenomenon of interest. We can thus say the following:

- We show how bounded rationality can be associated with the elements of the implemented co-evolutionary solution concept. This is done by finding reasonable explanations of the phenomenon in socio-economic literature, and the elements that are most likely to implement those explanations. We thus have a link. We associate bounded rationality with the representation (an element of the implemented co-evolutionary solution concept), given the explanation from one of the founders of bounded rationality research [133] in economics (the notion of a *trembling hand*), and recent work using this explanation [79, 122]. We use these explanations in socio-economic literature to define three types of bounded rationality: *implementation errors*, *perception errors*, and a *combination of the two*. This gives us a firm hold on the phenomenon of interest since we can indeed implement the representation in various ways, specifically with the aforementioned types of bounds.

- We then parametrise the associated elements (representation in our case) as this allows us to quantify the phenomenon of interest explicitly. We use two parameters, viz. *trembling probability (T)* and *randomness in moves (r)*, to model the three types of bounds within the representation. As such, we get a hold on the malleability of the phenomenon in explicit quantitative terms. Bounded rationality can thus be explicitly expressed and becomes, what we call, our reconciliation variable.

- We then define the characteristics of the implemented co-evolutionary solution concept that we are interested in understanding, i.e. evaluation metrics against which the role of bounded rationality within the simulations could be measured. Given the explicit quantitative implementation of the phenomenon, tangibly varying the phenomenon (i.e. varying T and r, given one

of the three types of bound), and analysing the relationship that the algorithm has with the phenomenon is thus made possible. This allows for the avoidance of mis-interpretations of the phenomenon from the bottom-up.

- We can see how bounded rationality affects the convergence characteristics of the implemented co-evolutionary solution concept, as opposed to leaving bounded rationality unchallenged and assuming its affects (as done in previous work [59]).

- We show how the three types of bounded rationality are related to the convergence characteristics and to each other, thus enabling a more methodical understanding of the algorithm in terms of bounded rationality, for its use as a model for socio-economic learning simulations in ACE.

Missing from ACE literature and thus a contribution, this exercise lets us discover the true algorithmic meaning of the phenomenon of interest, in addition to the way it affects co-evolutionary simulations. Note that this is in stark contrast to making unassessed assumption about the phenomenon. *Reasonably* (phenomenon having a backing in socio-economic literature – in our case it is literature on bounded rationality) *associating* (connecting the backed phenomenon with the algorithm), the phenomenon of interest with the algorithms, makes this exercise worthy of being conducted. Given this backing, one can at least be sure of the inner workings of the phenomenon, thus avoiding mis-interpretations of the phenomenon, and this can follow a systematic study.

Note that, it is not necessary to associate the phenomenon of interest with one particular element of the implemented co-evolutionary solution concept, nor it is necessary to use one particular explanation of the phenomenon from socio-economic literature. This methodology allows for reasonably associating socio-economic phenomena with one or more elements and analysing the effect that the changes in the parametrised and quantified elements (in turn, phenomena of interest) have on the envisaged goals from the simulation.

Final Statement About the Framework:

We thus provide a *holistic framework* for studying socio-economic phenomena of interest such that mis-interpretations of the phenomena be avoided altogether

from within socio-economic simulations. This framework is empowered by the analyses and refinements of simulations from the point of view of co-evolutionary solution concepts and reconciliation variables. We see this as one next step in ACE socio-economic learning simulation research, which must not be overlooked.

1.4 Thesis Outline

We now outline the structure of the thesis, giving a short summary for each Chapter that follows:

- **Chapter 2**: This Chapter covers the literature allowing us to assess the need for a bridge between the two fields viz. co-evolutionary algorithm design research and ACE, the latter when using co-evolutionary algorithms to model socio-economic learning and adaptation. We go through the history of economic games, in particular bargaining games, leading to understanding the need for computational approaches and socio-economic simulations therein. This is followed by covering research in the two fields mentioned above including, co-evolutionary algorithm design research, the use of co-evolutionary algorithms to simulate learning and adaptation in socio-economic games, the view of bounded rationality in economics and within these algorithms when used as simulations, the need for disciplining the use of these algorithms and, some significant efforts in the direction of disciplining the use of these algorithms. The Chapter ends with highlighting the missing link (which the thesis tries to then build in the following Chapters) between co-evolutionary algorithm design research and socio-economic learning simulation research within ACE.

- **Chapter 3**: This Chapter describes and applies the first part of the holistic framework that we propose for the analysis and refinements of co-evolutionary algorithms, specifically for a top-down avoidance of mis-interpretations of socio-economic phenomena within simulations. The Chapter thus promotes the idea behind co-evolutionary solution concepts from co-evolutionary algorithm design research, within ACE. A systematic study of previous work is carried out. This work mis-interprets the co-evolutionary

algorithm used therein in socio-economic terms, showing that a socio-economic phenomenon (bounded rationality) used for interpretations, explains the deviations of the algorithm from the envisaged goals (achieving convergence to the subgame perfect equilibrium solution in bargaining games). We study the work through the lenses of co-evolutionary solutions concepts. We see that we can indeed achieve the envisaged goals from the simulation by working on the implemented co-evolutionary solution concept, without using the socio-economic phenomenon for explanations. In effect, we challenge the interpretation of the socio-economic phenomenon in order for mis-interpretations of the phenomenon to be avoided, using the notion of co-evolutionary solution concepts. This allows us to understand what the socio-economic phenomenon does not mean. This further allows us to bridge the two fields and understand the link in greater detail by wanting to understand what the socio-economic phenomenon may mean within simulations, the methodology for which we then show in the next Chapter. Part of the work considered in this Chapter, in particular Sections 3.6.3.1 and 3.6.3.2, were published in [26].

- **Chapter 4**: This Chapter describes and applies the second part of the holistic framework that we propose for the analysis and refinements of co-evolutionary algorithms, specifically for a bottom-up avoidance of mis-interpretations of socio-economic phenomena within simulations. The Chapter thus proposes a methodology which allows for an explicit understanding of co-evolutionary simulations from the point of view of the socio-economic phenomena of interest under question. We show that we can reasonably associate socio-economic phenomena of interest with the elements of the implemented co-evolutionary solution concept. This allows us to parametrise and quantify the phenomena in explicit terms, by way of parametrising and quantifying the associated elements. We term the phenomena of interest thus quantified, as reconciliation variables. We then show how the relationship between the phenomena of interest and the algorithm (and thus the model of socio-economic learning and adaptation), can be understood in a systematic manner. Together with the first part of the framework,

this methodology relieves us from making implicit assumptions about the socio-economic phenomena of interest, which previously have been carried into the simulations unassessed and thus been mis-interpreted, specially phenomena which do not have an obvious connection with the algorithms (like bounded rationality, which has generally been assumed to be implicitly present in simulations). This helps make the co-evolutionary algorithm tangible, from the point of view of socio-economic phenomena of interest, such that the mis-interpretation of the phenomena be altogether avoided via this tangibility.

- **Chapter 5**: We conclude in this Chapter, elucidating the grasp on co-evolutionary algorithms that our framework, which is composed of the use of co-evolutionary solution concepts and reconciliation variables for analysis and refinements of these algorithms, offers to ACE. Thus we lay out the contributions of this thesis. We then cover further observations about our framework, specifically how it complements other work on socio-economic learning simulation research within ACE, and the weaknesses that still remain within the framework. We also cover future research directions our work enables foreseeing.

Chapter 2

Background

2.1 Introduction

In this Chapter we consider a review of the literature covering the history of socio-economic games, in particular bargaining games (Section 2.2), as we use these games as a test bed in order to study co-evolutionary algorithms, in particular for their usage in socio-economic simulations within ACE. A need for computational approaches for simulating learning in socio-economic games is highlighted. This follows a review of co-evolutionary algorithm design research (Section 2.3), where we highlight the issues that have marred co-evolutionary algorithms and the manner in which these have been understood and tackled, in particular using notion behind co-evolutionary solution concepts. These issues can seep into socio-economic simulations within ACE, as co-evolutionary algorithms are often used off-the-shelf for these simulations. We thus review their usage in ACE (Section 2.4), highlighting the possible breadth of application of these algorithms and their unaddressed issues, hence the affected application domains within ACE. We then review literature on the socio-economic phenomenon of bounded rationality, since the thesis takes this phenomenon as a case study to scrutinise the off-the-shelf usage of co-evolutionary algorithms (Section 2.5). Finally, we review the efforts that have been made in making the adoption of co-evolutionary algorithms more methodical within the agent based community (Section 2.6), specifically when using them for simulating socio-economic learning (the domain of application of

co-evolutionary algorithms within ACE that we are mainly concerned with in this thesis) and highlight the gap that still remains (Section 2.7). This gap is what we try to fill in the remainder of the thesis.

2.2 Bargaining

Bargaining, according to [135], and as the name suggests, is a socio-economic problem in which two individuals meet and cooperate towards the creation of a commonly desired surplus, where the actual distribution of the surplus puts them in conflict.

Suppose an individual wants to sell a commodity and another wants to buy it. Both have a common interest in the transfer of this commodity. Since there can be many prices at which the commodity can be sold, where the seller prefers a higher price and the buyer lower, conflict may arise. Note here that the monetary difference in what the seller wants and what the buyer wants to pay is termed as *surplus*. So, if the two do not cooperate, there would be no commodity transfer and hence, no surplus. In order to resolve the potential deadlock so as not to lose out on the joint gains (commodity at a lower price for the buyer and a good price for the commodity for the seller) that come about as a result of the commodity transfer, they have to find some way to reach an agreement on the price. Various social, economic and political scenarios which fit the definition are listed in [135], for example, a couple deciding on how to split intra-household chores, negotiations on a labour contract between a firm and a union, two unfriendly nations trying to reach a lasting peace agreement. Of more relevance to this research is a parallel scenario, with potential that may help understand and inform the above scenarios better, that of computational agents deciding on how much of a certain resource they might need. More on this theme will be elaborated on shortly.

According to [135], a bargaining problem essentially has three basic elements: an arrangement that is expected to take place when the involved parties are unable to reach an agreement (also called the disagreement point or the status-quo situation); the existence of mutual gains from cooperation; and the *multitude* of possible cooperative arrangements in case of an agreement leading to a split of the surplus.

2.2.1 Bargaining Before Game Theory

Bargaining problems were initially called bilateral monopolies and were deemed indeterminate [45] in orthodox economics [69]. It was argued in [150], that the most one can say about the problem was that the result (or solution) will lie in the bargaining set.

If a solution is in the bargaining set then it is said to be both *individually* and *collectively* rational. Individual rationality means that neither party should end up with pay-offs worse than what they get if an agreement is not reached (i.e. worse than the status-quo). Collective rationality refers to the concept of Pareto efficiency. An outcome is Pareto efficient if no outcome exists which is strictly preferred by one player and not less preferred by any other. Deviating from this outcome would make one player better off at the expense of the other. Pareto efficient outcomes are also referred to as outcomes lying on the contract curve [45] or the pareto-efficient frontier.

According to [69], Zeuthen attempted to provide a more determinate prediction (antedating the theory of games [150]) where the solution to the bargaining problem could be dictated by the parties' attitudes towards risk of a breakdown. If a party's readiness to risk a conflict (also termed as determination) is greater than the other, then the other party tends to make a concession. The process involves each party making a concession until no more concessions are possible due to the indivisibility of the monetary unit, setting a lower limit on the size of the admissible concessions. In the symmetric case (where the determination of each party is equal), an *equal division* of the bargaining surplus is obtained (a determinate solution!).

2.2.2 Assumptions in Game Theory and the Need for Computational Approaches

To make mathematical analysis possible, game theory often makes simplifying assumptions. Two of the most common assumptions made being players having *complete information* and them being *perfectly rational*.

Complete information refers to the fact that every player of the game knows the strategies and payoffs available to the other players. The players may not

however have knowledge inside the game. If they do then the game is thought of being that of *perfect information*. For example, in the case of prisoner's dilemma, complete information about the strategies and the payoffs available to the players may be known but the players do not necessarily know the moves of the opponent in any particular round of the game (or in the first round, depending on it being a one shot or a repeated game). This lack of information or *uncertainty* inside the game makes it a game of *imperfect information*. On the contrary, a sequential game (for example, the bargaining game that we consider in this thesis in the Chapters to follow) where, at each decision point, the choices that have previously been made are known, is said to be that of perfect information.

A game is that of incomplete information if something about the circumstances in which the game is being played is not known to the players [58]. For example, the utility functions (or preferences) of the players (in the bargaining context) may not be known. In such a game however, the players may be forced to consider an infinite (because any piece of information has infinite hierarchical levels of being known, in terms of knowledge of the knowledge ad infinitum) hierarchy of beliefs[1] [18, 70]. Incomplete information of other player's preferences and beliefs is modelled by game theorists by specifying a limited number of player types [58, 70]. Types essentially determine the preferences and beliefs uniquely. Players are not certain about the opponent's type but the probability that the opponent is of a certain type, i.e. probability of a type of the opponent conditioned on the opponent, is *common knowledge*[2]. The game is then said to be transformed into that of imperfect information. Imperfect information games are useful for studying phenomenon like reputation building [128]. The assumption of perfect rationality follows from the assumption of common knowledge on how players reason.

These assumptions limit the practical applicability of game theoretic solutions. Moreover, real life trading situations do not assume complete information and perfect rationality. Both humans and computational agents have limited information and forward looking capabilities. As stated in [58], many tasks are

[1]Beliefs are probabilities of events happening about which the player is uncertain.

[2]Knowing information about players, and the players knowing that the information is known, and knowing that the players know that the information is known, ad infinitum, describes the notion of common knowledge.

learnt through a process of trial and error (i.e. through experience). Additionally, agents may be programmed by different parties implying them having different capabilities. Assuming perfect rationality about their behaviours will thus be erroneous. Indeed, an agent *"satisfices"* [138] (see Section 2.5), given the information it has, when it has it.

According to [129], if an agent has little a-priori knowledge about the environment and it gradually adapts in the quest for an optimal solution (by interacting with its environment through a process of trial and error), it can be said to be *boundedly rational* [60, 133, 138] (see Section 2.5).

In a computational scenario with intelligent agents, due to the advent of new computational modelling tools, the limiting game theoretic assumptions of full rationality and complete information are unnecessary and not required anymore. This is because the behaviour of agents can be modelled directly [58] in their *"full dynamic capacity"* [145]. Agents can use machine learning techniques to improve on their strategies (after being put into the environment with a certain strategy). After a period of learning, the agents may be able to exhibit behaviours resembling rational and fully informed agents.

Considering the above, computational techniques are thus a positive way forward and may have many advantages, two of which, for the kind of application of these techniques that this thesis is concerned with, being: the ability to be used for understanding game play in realistic environments and the ability to analyse game theoretic settings which are too complex to be analysed analytically, thus cumbersome or even intractable.

2.2.3 Game Theoretic Approaches to Bargaining

There are essentially two branches of bargaining theory and game theory in general: *cooperative* or *axiomatic* and *non-cooperative* or *strategic*.

In *cooperative game theory*, pre-play communication is allowed and the players can co-operate into reaching binding agreements. Given a set of feasible outcomes (the bargaining set) which are the outcomes that can be jointly achieved by players concerned, finding a solution from within this set without calling into question how the players reach it, is the main idea behind the cooperative approach. A

number of properties (or *axioms* or assumptions) that the solution to the bargaining problem should have are proposed and the outcome which best agrees with these properties is chosen as the solution. This theory was born with Nash [101]. The theory answers the question of how bargaining should be resolved between rational parties in accordance with some desirable principles [135].

The *non-cooperative approach*, initiated by Nash [103, 104] relies on the exact specification of the situation under study (bargaining or negotiation in the current context) as games and the identification of the behaviours occurring in these games. Here, an exact specification signifies specification of the protocol using which the players interact, the strategies the players can use, the payoffs they can get, the information available etc. This approach to game theoretic problems and bargaining in particular describes how the bargaining process may evolve or proceed in the presence of common knowledge of rationality ([135]).

An overview of the bargaining literature from these two branches of game theory is considered next.

2.2.3.1 Cooperative/Axiomatic Approach

Nash [101] was the first to contribute to the axiomatic theory of bargaining. A bargaining problem (more specifically, a Nash bargaining problem) is represented by a pair (S, d) in utility[1] space where S is a closed, bounded above and convex subset of $\mathbb{R}^2$ and is the set of feasible[2] utility pairs, and d is the point in S representing the payoffs that players get in case of a disagreement. Accordingly, a solution (also referred to as a **solution concept**[3]) to the bargaining problem is a function that maps a feasible set of utility pairs to one of its feasible points, i.e. $F : (S, d) \rightarrow s$, where the utility pair $s \in S$, in the above context. Note that the nature of S dictates the validity of a solution concept.

An example of a solution concept is the disagreement solution [135] which assigns the point d to the bargaining problem and is rather pessimistic (as evident). This solution is not pareto-efficient as it does not exploit the gains from reaching

[1] Utility can also be seen as the payoff a party or player gets.

[2] A point is feasible if it is possible to select it, in the sense that, it can represent a feasible pair of payoffs that may result from a game between two players.

[3] See Glossary.

a cooperative agreement. The dictatorial solution [88], where one bargainer acts as a dictator, assigns the point in the bargaining set where the other bargainer receives zero utility as the solution to the problem. As can be seen, this solution seems unfair.

Nash [101] proposed four desirable properties that a bargaining solution should possess:

1. The final outcome does not depend on how the utility scales are calibrated as different utility functions can be used to model the same preferences.

2. The agreed payoff pair should be in the bargaining set (individually and collectively rational).

3. Certain utility pairs are irrelevant in that if players agree on a pair s when t is also feasible, then t is never agreed on when s is feasible. The decision on choosing s does not depend on t whether or not t is present in the feasible region. This axiom is also called *independence of irrelevant alternatives (IIA)*.

4. Both players get the same in symmetric situations (if the player's labels are reversed, each will still receive the same payoff).

The solution proposed by Nash [101], satisfying the four properties above, is the utility pair $s = (u_1, u_2)$, where u_1 and u_2 are utilities or payoffs of the players, which maximises the Nash product $(u_1 - d_1)(u_2 - d_2)$, where $d = (d_1, d_2)$[1] and this solution is unique [101]. If one relaxes the symmetry axiom, the bargaining powers (does not mean bargaining skills due to the assumption of perfect rationality, but possibly, strength due to market positions for instance) of players dictates the solution (according to [19], this is also called the generalised Nash bargaining solution). It is the payoff pair $s = (u_1, u_2)$ which maximises the product $(u_1 - d_1)^{\alpha}(u_2 - d_2)^{\beta}$, where α and β are the bargaining powers of the players.

Attitudes of bargainers towards the risk of a breakdown or disagreement also effects the payoff that they receive if one follows Nash's solution concept. Consider

[1] d_1 and d_2 are the payoffs that the players get on a disagreement. They could well be zero, in which case the Nash product is simply $u_1 u_2$.

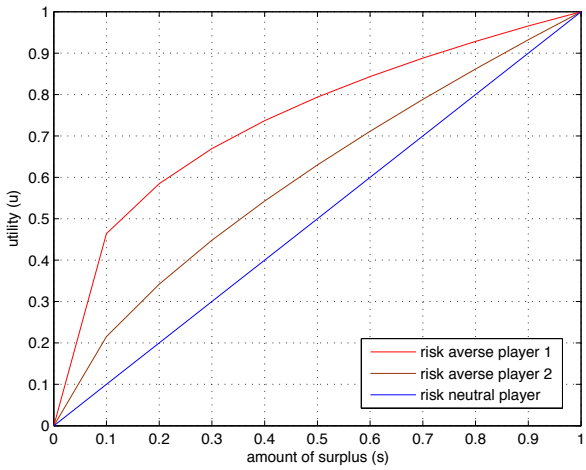


Figure 2.1: Possible utility functions for risk averse and risk neutral bargainers.

the utility functions in Figure 2.1. These are of the form $u(s) = s^{\alpha}$, where $0 < \alpha \leq 1$ and the smaller the α, the more risk averse the bargainer is (risk averse player 1 is more risk averse than risk averse player 2 in the figure). In the symmetric case (when the bargaining powers are equal), a more risk averse player gets a lower portion of the surplus, as shown in [19] and [135]. A risk neutral bargainer ($\alpha = 1$) will get a higher share of the surplus when confronting either of the other two players shown in the figure.

Nash's IIA axiom was the subject of severe criticism [92]. As per [135], Nash's solution does not consider global issues (in terms of the bargaining situation at hand) such as the highest utility each bargainer can obtain. An alternative solution to the Nash's bargaining solution was proposed in [83]. They define the notion of having a *utopia point*. If $a_i(S)$ be the highest utility player i can achieve (also called aspiration level) then the utopia point is the point in $\mathbb{R}^2$ (typically not feasible) denoting the aspiration levels of each player $(a_1(S), a_2(S))$. The Kalai-Smorodinsky solution thus selects the maximum element on the line that joins the disagreement point (d_1, d_2) with the utopia point $(a_1(S), a_2(S))$ and in the bargaining set. Other solution concepts replacing the IIA axiom can be found in [125] and [146].

Another solution concept is that of egalitarianism [100]. An egalitarian solu-

tion is a point on the pareto-efficient frontier where utilities for the players are equal. This solution is more tied to ethical behaviour than to principles governing bargaining between rational individuals. Also, if a bargaining solution maximises the sum of the utilities, it is said to be a utilitarian solution [100]. Due to interpersonal comparisons of utilities being possible in both egalitarian and utilitarian solution concepts, Nash's axiom of solutions being independent of utility calibrations does not hold. Cooperative theories of bargaining are discussed in greater detail in [126].

Traditionally, bargaining problems have involved parties trying to reach an agreement over a single issue e.g. the division of a surplus (price of a commodity for instance). Such bargaining situations are termed as distributive [15] in that a gain for one player always creates a loss for the other. The above mentioned solution concepts were initially developed for single issue bargaining problems.

Often though, individuals have several goals which they want to achieve, motivated by self-interest. This, in economic literature, is termed as individuals having multiple issues to deal with (e.g. price of a commodity, delivery time, quantity etc.). A very important point with multi-issue bargaining situations is that parties have preference relations between issues such that when dealing with another party, a mutually beneficial outcome can be agreed upon by exploring the trade-offs that the issues present. As opposed to distributive bargaining, it becomes possible for one side to gain without the other side getting less. Such negotiation scenarios are termed integrative [15]. Moreover, due to the existence of trade-off agreements, multi-issue bargaining is less competitive as opposed to single issue scenarios which are often also termed as competitive negotiations [68].

In the case where multiple issues are involved, these concepts can still be applied provided we can transform the combined utility space (each issue occupying one dimension in the space) into a single issue utility space. This can be done by utilising a multi-attribute utility function [58, 120] [1]. This utility mapping is appropriate only if the issues are independent (i.e. contribution of one issue is

[1] A multi-attribute utility function defines utility over multiple weighted attributes (or issues). The relationships or trade-offs between issues are assumed given in this case and are the weights assigned to each issue. The weights indicate preferences or the relative importance one gives to the issues. The utility is thus calculated by multiplying the outcome on each attribute (outcome on each issue) with the weight associated with it and adding across all the issues.

independent of the values of the others). Once the preferences of the bargainers are mapped onto the multi-attribute utility function, the problem reduces to that of a single issue. Examples of such problems can be found in [120].

2.2.3.2 Non-cooperative/Strategic Approach

As mentioned previously, details of the negotiation process are specified in this case. There are various protocols (which may be called games) that researchers have used to specify the bargaining process. The idea is that bargainers will use these protocols while deciding on the division of the surplus and some behaviour will emerge. The goal is to study procedures that are reasonable (specify the interaction procedure or protocol in a simplistic yet realistic manner) and identify rational behaviour within them. To determine rational outcomes of a game, the concept of equilibrium is used in this approach. The two most widely used equilibrium concepts are *Nash equilibrium* due to [102] and *Subgame perfect equilibrium* due to [133]:

Nash Equilibrium: If no player can benefit by unilaterally changing its strategy, then the strategies chosen by all players are said to be in a Nash equilibrium. Every finite game has at least one such equilibrium point ([102, 103]).

A game is said to be one of extensive form if it can be represented by a tree and the players can make decisions sequentially at various stages or rounds of the game (at nodes of the tree which are also called decision points).

Subgame Perfect Equilibrium: Considering an extensive form game, the strategies are said to be in subgame perfect equilibrium if they constitute a Nash equilibrium at every decision point.

To illustrate the difference between Nash equilibrium and subgame perfect equilibrium, let us consider the simple version of a two round Chain-Store game [134] with the game tree structure and the payoff matrix as shown in Figure 2.2. The tree illustrates the extensive form version of the game, whereas the matrix shows the normal form version. The idea is that there is an incumbent firm having a monopoly in a market. It is threatened by a new entrant firm which may or may not enter (IN and OUT moves). In case the entrant enters, the options for the incumbent are to FIGHT or NOT FIGHT the entrant. The payoffs are

as shown in Figure 2.2. It can be seen that, in the normal form version, if the incumbent chooses to NOT FIGHT, the rational choice for the entrant is IN (and vice versa, if the entrant chooses IN, the rational choice for the incumbent is to NOT FIGHT), thus NOT FIGHT and IN are one Nash equilibrium. There is another Nash equilibrium, that where, if the incumbent chooses to FIGHT, the rational thing to do for the entrant is to stay OUT (and vice versa, if the entrant stays OUT, the incumbent will have to be choosing or intending to choose FIGHT, otherwise the entrant will choose to get IN). So, there are two Nash equilibria in the game. One of these however is more attractive if we expand the game into its extensive form. Taking this view, the players move sequentially with the entrant choosing their move first and the incumbent choosing once the entrant has made a choice. Fully rational players will have the tree laid out in front of them with the respective payoffs, from the very start. If so, the entrant will know that the only way it is getting a positive payoff once the incumbent has made its choice in the next round, is by entering (IN) and that the incumbent will have to choose to NOT FIGHT, otherwise it will get a negative payoff. Thus, there is an equilibrium where the entrant chooses to go IN and the incumbent choose to NOT FIGHT. Taking the sub tree depicted in the Figure, which is known as the subgame, the best the incumbent can do is to NOT FIGHT, and this happens given the entrant choose to move IN. Thus, this subgame has an equilibrium too. We can see that in the extensive form version of the game, every subgame (including the full game tree) has the same Nash (because it satisfies the definition of no player unilaterally choosing to play differently) equilibrium. This is the subgame perfect equilibrium. The process of looking at the entire tree and working out the best moves, given the opponents best moves from the last to the first round is called backward induction and we will see later how this can be used for the bargaining case.

We now consider some games studied in strategic bargaining literature. The first bargaining model to be considered as a non-cooperative game was introduced in [104]. This is called the *Nash demand game*. The idea is that two players simultaneously demand a utility level or a part of the surplus. There is no knowledge of the other player's demand. If the sum of the demands exceeds the surplus, the players receive their disagreement payoff otherwise they get their respective parts.

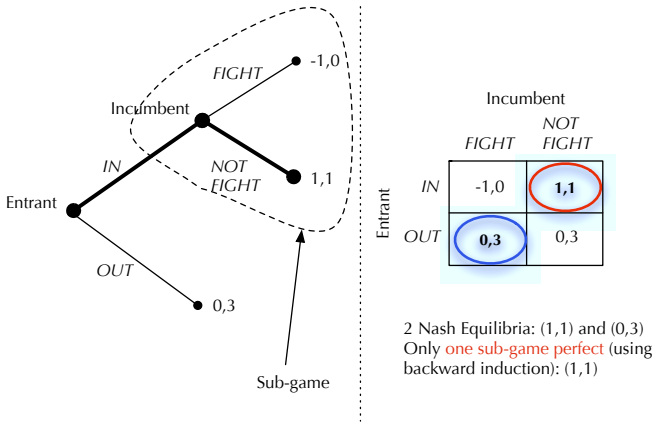


Figure 2.2: Example of the simple two round Chain-Store game [134] showing the difference between Nash Equilibrium and Subgame Perfect Equilibrium.

This game admits an infinite number of Nash equilibria. The outcomes on the pareto-efficient frontier (i.e. collectively rational payoff pairs) are all equilibrium points. Moreover, the disagreement point is also a Nash equilibrium. Suppose the players chose a point on the pareto-frontier. It can be easily seen that any player would only decrease their payoff by moving away from the frontier. Also, if both demand more than the entire surplus, they will get the disagreement payoff even if they unilaterally want change their strategy.

Another game, considered less competitive and more generous from the point of view of throwing away of the surplus (which may easily happen in the Nash demand game), is the *Ultimatum game* [19]. It provides a model for the simplest of all bargaining procedures and is realistic in the sense, it is the bargaining procedure normally employed by us in stores [19]. This is a one or two round game in which the second round is similar to the Nash demand game. In the first round, one player proposes a split of the surplus and the other player has to decide on whether to accept or reject the offer. In case the offer is rejected, both players either get nothing or the Nash demand game is played. The usual definition says that the game ends in case the responder rejects the offer and both players get nothing [19]. This game has a subgame (trivially, the entire game) and has a subgame perfect equilibrium in the event where one player demands

the whole surplus and the second accepts this deal. The game again admits a continuum of Nash equilibrium outcomes (the pareto efficient outcomes).

One of the most elegant protocols [58] extends the aforementioned ultimatum game to multiple rounds. It models negotiations as they take place over time and is referred to as the *alternating offers game*. This game is considered throughout the thesis. As mentioned in [58] and [135], work on this bargaining model has been pioneered by Stahl. The game can be characterised as being that of either a finite horizon (or finite number of rounds/periods) or an infinite horizon (studied by Rubinstein [127]). Assuming the surplus is of size 1, in period 0, Player 1 starts by making a proposal (a division of the surplus, proposing x for himself). Player 2 can either accept or reject the proposal. If he accepts, Player 1 gets x and Player 2, $1-x$. If he rejects, they meet again at the negotiation table after a period of time and then (period 1) the roles of the players switch. Now Player 2 makes a counter offer, which Player 1 then accepts or rejects (sending the game to the next round). In the finite horizon game, there is a finite number of rounds i.e. there is a deadline. If an agreement has not been reached until the deadline, both players receive zero payoff. In the infinite horizon game there is always a new proposal in the follow up periods until an agreement is reached. *Starting at the last stage of the game and working backwards inductively leads to the identification of optimal strategies for rational players with perfect information.* This approach is similar to dynamic programming and is also the idea behind finding equilibrium strategies which are optimal responses at every point in the game and not just at the beginning of the game for all players concerned. The strategies thus obtained are in subgame perfect equilibrium (SPE). To predict optimal outcomes in dynamic games such as the alternating offers game, one often comes across the term *discount factor*. A discount factor models the utility of a future outcome as present valuations as expectations. In other words, a player values future events but not as much as present events, but not because he does not appreciate the future event happening, instead, because he may be unsure of them happening. In effect, a discount factor models how impatient a player is.

To give an idea of how this backward induction mechanism and discount factors work, lets consider a two stage alternating offers game. Suppose the surplus size is 1 and the utility functions are linear. We know that SPE in an

ultimatum game is $(1, 0)$, where the first player gets the whole of the surplus leaving the second with nothing. Suppose, in the two round game, a node is reached where Player 2 has to make an offer. This round on is a subgame of the actual game and can be seen as an ultimatum game by Player 2. In this case, Player 2 will play his SPE strategy and offer $(0, 1)$, i.e., full surplus for himself. The expected payoff at this node is dictated by the discount factors of both players (say δ_1 and δ_2). The expected utilities will thus be $(0.\delta_1, 1.\delta_2) = (0, \delta_2)$. Now, if we traverse backwards into the previous round of the game, Player 1's optimal move, with what is expected in the next round known, would be to make an offer which makes Player 2 indifferent between whether to accept or reject. This offer is $(1 - \delta_2, \delta_2)$. In a way, Player 1 convinces Player 2 not to take the entire surplus (in case the negotiations go to the second round), by offering him the present discounted value of the entire surplus. Note that if Player 1 offers anything lower than δ_2 to Player 2, he will end up with nothing as Player 2 will reject the offer and play his equilibrium strategy with the counter offer of $(0, \delta_2)$. Also, rationality will make Player 1 not to offer Player 2 anything more than δ_2. The offer $(1 - \delta_2, \delta_2)$, as it will result in δ_2 for Player 2, will be accepted by him, as by doing so, he will be playing his rational move. The strategies resulting in the outcome $(1 - \delta_2, \delta_2)$ are in subgame perfect equilibrium as no player will benefit from unilaterally changing his strategy at any round of the game i.e., the strategy is a Nash equilibrium of every subgame of the two round game. We can also see that if Player 2 is very impatient, δ_2 will be nearly zero and in this case, Player 1 will get nearly all the surplus. On the other hand, if δ_2 is nearly one, Player 2 will get the major share. Also, the agreement takes place immediately. Such a backward induction process is usually used to determine the outcome of rational play in dynamic games such as this. Similar logic applies and continues to any number of rounds.

Rubenstein [127] demonstrated that Nash equilibrium does not restrict the outcomes i.e. a unique solution to the alternating offers problem cannot be found using the Nash equilibrium concept. In fact, any agreement $(x, 1 - x)$, in any period t of the game where $0 \leq t < \infty$, can be supported as a Nash equilibrium outcome [127]. However, the concept of SPE was applied by [127] showing that there exists a unique SPE in the alternating offers bargaining model specified by

the partitioning where Player 1 gets $(1-\delta_2)/(1-\delta_1\delta_2)$ (δ_1 and δ_2 being fixed discounting factors of the players) and Player 2 gets the remainder of the surplus immediately (in the first round of the game). The time period between two stages of the game is usually not taken to be unity. If τ is the period between two stages, then δ_i is replaced by δ_i^τ everywhere. An important point to make here is that *for small time intervals between rounds i.e., if* $\tau \to 0 \Rightarrow \delta \to 1$, *this unique SPE outcome approximates the generalised Nash bargaining solution.* This important result was convincingly proven in [20]. The limiting case of $\tau \to 0$ is significant in that, in real world situations, the optimal thing to do for a negotiator, once he rejects an offer, is to make a counter offer as soon as possible without sticking to a strict timetable [19]. This result essentially established a link between cooperative and non-cooperative game theory and is called the Nash Program [19]. As an aside, note that using computational methods to establish a link between cooperative and non-cooperative game theory is an interesting concept and may be seen as a Nash program too, but this is just our conjecture, and to our knowledge, not been seen in this light before. Whether it is interesting or not is beyond the scope of this thesis to discuss.

The Rubinstein-Stahl alternating offers game, as this alternating offers protocol is called, has had many applications in real world situations e.g. bargaining problems concerning international trade, industrial organisation, political economy etc. [135]. More on infinite horizon games and its applications to decentralised trading situations is given in [110]. The prevalence of this protocol is one motivation for us to consider it in this thesis.

Yet another protocol studied in the non-cooperative bargaining literature is the *monotonic concession protocol* [123]. Two players simultaneously announce their proposals and an agreement is reached if both offers match or the offer exceeds the other player's demand. Tossing a coin decides on which one of the two offers to chose from, in case they are dissimilar. In case of a disagreement, the players make new offers which need to be higher in utility for the other player than the utility he (the other player) would get from the previous offer, implying monotonic concession. The player can either make the same offer in which case he is said to be standing firm, or could concede. If both players stand firm, negotiations end and both receive a disagreement payoff. More on this protocol

can be found in [123].

According to [119], in the multi-issue case, the protocols under which bargaining may take place, can be divided into global or simultaneous, separate and sequential types. As the name suggests, global bargaining calls for having to bargain over all the issues simultaneously while separate bargaining allows for negotiations to take place over issues separately and independently. The third case involves negotiations taking place over issues in a sequential manner. Sequential bargaining can further be divided into independent and simultaneous, depending on whether or not players can benefit from an agreement over an issue. In the independent implementation, an agreement over an issue excludes them from being considered in future bargaining rounds and the agreed upon issues are no longer discounted. On the contrary, the simultaneous version makes players wait until all issues have been considered before benefits from them can be enjoyed. As may be evident, if one negotiates over issues in a sequential manner, the question of which issues one should consider bargaining over first arises. There are two main bodies of work dealing with this issue. The order in which different issues are brought on to the negotiation table (also called agenda) can be set either exogenously or endogenously by the players. In the former case, since each issue is bargained over one at a time, Rubinstein's results of uniqueness of the equilibrium and that of it being pareto-efficient hold [135]. Fershtman [51, 52] studies exogenous agenda games in the case where the players attach different importance to different issues. Usually however, either the importance associated with issues is equal (for instance, see [14]) or the players have identical preferences [24] (i.e. a player attaches the same importance to an issue as the other). More realistic games with players having to endogenously decide on the agenda have been studied in [78] and [77]. As per [77], offers can be made in any subset of remaining issues and a unique subgame perfect equilibrium agreement is shown to have been obtained.

A case in point to make here is that the reason having a unique subgame perfect equilibrium for a game is important is that such games become easier to have as validation benchmarks, i.e. it becomes easier to test new approaches (from outside game theory) to solutions to these games by comparing the results with game theoretic ones (i.e with equilibrium outcomes). We elaborate on this

point in Chapter 3.

2.2.4 Computational Approaches to Bargaining

Modelling human socio-economic learning behaviour using computational approaches necessitates getting rid of the notion of having to make simplifying assumptions of rationality and common knowledge used frequently in game theoretic analysis. Techniques from artificial intelligence have been shown to attempt modelling such behaviours to good effect and rational behaviours have often been shown to emerge. However, these techniques have many issues, which we will discuss in Section 2.3, specifically for one popular technique considered throughout this thesis: *co-evolutionary algorithms*.

The idea with computational approaches is to make the agents interact with each other according to some interaction protocol and let them, in time, learn from these interactions. An agent can learn the bargaining strategy specifying the actions it should take during the course of play while adapting the strategies over interactions. In other words, an agent can learn and see how to modify its strategy based on previously played bargaining games and try to amend previous mistakes as necessary. The agent may modify its strategy on the fly (whilst in a game) at the same time as well. The latter situation may arise if the agent is unsure of the opponents preferences or type and tries to form beliefs letting it fine tune its behaviour.

Genetic algorithms have been used [107] to learn negotiation[1] strategies in both single and multiple issue bargaining settings. Agent strategies were represented as binary coded strings which encode the threshold (determining whether an offer should be accepted or not) and a counter offer (in case of rejection of the offer by the opponent and if the deadline has still not been reached). A similar model is studied in [58, 59], and motivates much of the work in this thesis, a detailed discussion of which is considered in Chapter 3. It encodes the strategies as a string of real numbers. The evolution of negotiation strategies where the offers and counter offers are generated by combining tactical information encoded in the strategies was studied in [97]. Two populations (buyer and seller) compete

[1]Note that negotiation and bargaining are used interchangeably in the thesis.

with each other and the strategies are co-evolved.

Apart from learning bargaining behaviours, techniques from machine learning such as bayesian belief models have also been used to figure out the opponent's type [87] and thresholds of offer acceptabilities [160]. The problem with using beliefs to adapt agent strategies is that agents should be able to form beliefs of opponents beliefs and this belief structure continues in an infinite regress. As pointed out earlier (Section 2.2.2), [70] suggested limiting the number of player types with the preferences of each type being common knowledge to get around this problem of infinite regress, making mathematical analysis possible. However, when using computational agents, the notion of bounded rationality and limited reasoning relieves us from having to know the beliefs of opponent's beliefs and so on. The use of kernel density estimation to approximate the preferences of the opponent (in the multi-issue bargaining case) in order to let the agents make more effective negotiation trade-offs, is investigated in [34].

We are essentially concerned with co-evolutionary algorithms in this thesis and the manner in which they have been applied or adopted, specifically as modelling socio-economic learning (as one of the things they are meant to be doing in ACE). We choose bargaining games to explore and understand issues, and propose solutions to the issues, with the application of these algorithms. There is a gap in understanding co-evolutionary algorithms when used for socio-economic simulations. There are problems with these algorithms which are generally overlooked within ACE, causing those problems to transfer into socio-economic simulations. We consider co-evolutionary algorithm design research, highlighting problems with these methods in Section 2.3. This is followed by looking at the use of co-evolutionary algorithms in socio-economic simulations (Section 2.4) and significant efforts in the literature trying to bridge the gap (Section 2.6). This thesis is yet another effort in this direction, as the Chapters to follow will reveal.

2.3 Co-evolutionary Algorithms

Co-evolution is the process of adaptation, that occurs amongst agents interacting with each other in some manner, in turn revealing a reward structure, towards increasingly complex or refined behaviours. Co-evolutionary algorithms were in-

troduced into the field of evolutionary computation as an alternative to traditional evolutionary optimisation methods, which were based on optimising with an explicitly defined fitness function. On the contrary the fitness of an individual in co-evolutionary algorithms is evaluated and depends on other individuals co-evolving with it. Of much influence was the work in [72]. The problem that co-evolution tackled there was the problem of optimising sorting networks. Co-evolutionary algorithm were deemed superior to genetic algorithms (one type of evolutionary algorithm), the latter based on a hand designed fitness function. Co-evolution was used in this work, i.e. in [72], as an alternative to evolutionary methods, for search efficiency reasons. The nature of the problem did not impose using co-evolutionary algorithms. A static function could have been designed by hand, i.e. the fitness landscape does have a structure that allows for evaluating and comparing solution performances, yet co-evolution results in better search. The interaction between opponents to choose from for fitness evaluation was also seen important for better performance of co-evolution over standard evolutionary methods.

The view of co-evolution as an optimising mechanism with advantages over standard evolutionary methods was generally understood as being caused by the popular *"arms race"* hypothesis. Taking the view of host-parasite interactions [124] for purposes of explaining what this means, the general idea was that, the success of the host against parasites depends on the host out-competing the parasite in some manner. This results in new challenges for the parasite to overcome. As the parasite evolves, upon tackling the challenges, this creates new challenges for the host to overcome. Thus, the challenges are continuously created and bounced across from one side to the other. This may lead to an arms race, and thus creates an environment where increasingly complex or refined behaviours may result.

As suggested in [106] and further pointed out in [98], *"competing populations may reciprocally drive one another to increasing levels of complexity by producing an evolutionary arms race"*. However, it is also pointed out in [98], that arms races do not imply an increase in complexity, but instead they imply mutual efficiency or refined behaviours between lineages. Nevertheless, in general, the advantages that co-evolutionary algorithms have over standard evolutionary

methods has been seen as coming from co-evolution creating an arms race, co-evolving individuals having to deal with a variety of opponents, making general behaviours appear, and co-evolution possibly preventing stagnation within local optima. These advantages are not all free from their own specific pitfalls however. Gradual changes in the selection environment that creates a shift in challenges from mild to increasingly difficult (arms race) can, in fact, lead to one population *"utterly"* [98] defeating the other, which leads to *"disengagement"* or *"loss of gradient"* [154]. This, as pointed out in [98], would be similar to an *"extinction event"* in nature. With regard to generalisation giving solutions that are capable of out-competing a wide variety of opponents, a standard search problem may not necessitate this, as there is only one problem to solve. Then, it really depends on what one wants from co-evolution. This may result in diversity in the population that further aids search, but it may as well lead to phenomenon known as mediocre stable states [118] or collusion. This can be seen as a form of dynamic local optima where the population may just cycle around the search space. The issue of generalisation has recently been considered theoretically so as to understand how co-evolution can lead to solutions that, on an average, will be better at coping with a variety of opponents [28]. The goal there [28] is to study co-evolution and define a progress measure in terms of generalisation, rather than use co-evolution as an optimisation tool to solve a particular problem.

Arguments above lend a view of co-evolution that puts their use as simply an alternative to standard evolutionary algorithms into question. Co-evolution has however also been used to study and simulate the open-ended emergence of behaviours [140], reduction of domain knowledge required to learn complex control [90] etc. At the same time, games have been used as test beds to understand the co-evolutionary process properly [28, 36]. In general, these arguments also suggest that co-evolution has problems and that it is an engineered approach to optimisation and indeed an engineered simulation tool, if used as the latter. Further elaborating on co-evolution being an engineered product for artificial simulations, a look at the interaction and fitness evaluation detail is taken into consideration in [98]. As suggested [98], co-evolutionary algorithms developed so far tend to offer trade-offs (i.e. they are engineered to offer trade-offs) between *"fairness"*, *"generality"* and *"computational cost"*. Fairness here means considering evaluating

individuals against opponents which offer the same level of difficulty. Generality suggests diversity in the competitive environment within which an individual is evaluated. Computational cost suggests the cost of evaluating against a group (of a certain size) of opponents.

Note that, already, it is becoming clear that co-evolutionary algorithms are themselves marred with problems. Imagining their use, and that too off-the-shelf, for socio-economic simulation, as we will consider in detail in Section 2.4, does not relieve co-evolution and thus socio-economic simulations from problems. The need for a methodical approach to understanding co-evolution will only help in understanding the use of it as a simulation tool in applications of interest. As such, this understanding has a potential to make their application methodical too. Note that co-evolutionary algorithms considered in this thesis fall under the category of *competitive co-evolution*, the other category being *cooperative co-evolution* [156]. The main difference between the two is that the latter involves the co-evolution of parts of the solution to the problem being tackled, such that these parts are eventually brought together into forming the solution.

We now consider research into the pitfalls in co-evolutionary algorithms to highlight some efforts towards discovering and addressing them. This is followed by elaborating on the idea behind *co-evolutionary solution concepts*, which we use as a design tool in this thesis, to understand and address issues with co-evolutionary simulations.

The problems discovered in co-evolution essentially spin out of the lack of a method in designing these algorithms. It is not hard to see why these algorithms were then chosen off-the-shelf for socio-econonmic simulation, and really emphasises the lack of knowledge exchange between the two fields: co-evolutionary algorithm design research and socio-economic simulation research (more specifically agent based computational economics research, a detailed discussion of which is carried out in Section 2.4). Essentially, the problems in co-evolution can be listed as follows:

- **Intransitivity:** This suggests that the nature of the global fitness landscape, which is unknown apriori, may be such that, given three individuals X, Y and Z, if X is superior to Y, and Y is superior to Z, it does not necessarily follow that X will be superior to Z. Essentially, X may have evolved

(locally or temporarily i.e. without having any idea about Z) against Y, and Y may have evolved locally against Z, the adaptations that make X defeat Y may not make X defeat Z. The way superiority is defined is thus called into question and implies local progress. This phenomenon has been called by many names in the literature viz. *"cycles"* [106], *"relativism"* [154], *"mediocre stable states"* [118], *"collusion"* [118] and *"red queen dynamics"* [112].

- **Loss of gradient:** Out-competing opponents utterly is the suggested idea here [154]. The gradient is lost if one population *"utterly"* defeats the opponent population i.e. any individual in the winning population is able to *"utterly"* defeat any individual in the opponent population. The gradient may still be there in case of real valued scores from interactions, but the defeated population may be *"hopelessly maladaptive"* [98].

- **Parasitism:** Reaping *"easy gains"* [154] against opponents by exploiting specific weaknesses can allow primitive individuals to thrive at the expense of more sophisticated opponents [141], and can lead to *"displacement"* of promising opponents [98].

- **Forgetting:** The loss of certain characteristics or *"traits"* [53] from the entire population of individuals such that these traits may have resulted in a gain in fitness at some later point in time during co-evolution, is termed forgetting. This is also known as *"focusing"* [154] if the loss is due to drift caused by the biases in the variation operators, and the selection pressure becomes too narrow.

- **Fitness deception:** The nature of the search problem may result in certain solutions being more prevalent in the population and depending on the type of interaction mechanism used to calculate fitness, reduce the value of the more optimal solution, thus resulting in what is known as fitness deception [39, 54].

A generic remedy for these pathologies [53] in co-evolution is that of maintaing diversity at both the genotypic and phenotypic level, as mentioned in [53], and

there are various ways diversity has been injected. These include, halting co-evolution in one population [109, 113], spacial populations [72, 111], multiple isolated (in the sense reproduction across populations is prohibited) populations [23, 76], and speciation techniques [36, 37, 81, 124].

In general, there is a major problem echoed across co-evolutionary algorithm design literature that co-evolution does not do what is expected of it. The discrepancy between expected and actual outcomes from co-evolution is not just because of technical limitations in the algorithm [98], but also due to the fundamental nature of the search problem being addressed by co-evolution, which is complex enough to have lead previous work in co-evolutionary algorithm design research to have made unassessed assumptions. Essentially, assuming that progress towards an expected outcome will happen if an *intuitive design* of the algorithm suggests so, has been shown to be clearly wrong. The notion of progress is broken down into three types in [98], viz. local, historical and global, in order to elaborate on these assumptions. It is noted in [98], that there were two implicit assumptions in early co-evolutionary algorithm design research, namely:

- **Local progress implies historical progress** [98]: The assumption here is that an arms race is expected from co-evolution if new individuals are *"superior"* to their ancestors. The meaning of superiority is non-trivial in co-evolution however, as there is no fixed fitness function that can be used to measure the performance of an individual. This fitness function needs proper definition and is linked with the other parts of the algorithm. Cycles seen in co-evolution [106] and work on *"tracking the red queen"* [33], as mentioned in [98], are early works that reveal the problem with this assumption.

- **Historical progress implies global progress** [98]: This is to do with the fact that an arms race does not imply optimality. As discussed in [98], historical progress has often been seen as the kind of progress one needs in co-evolution. More precisely, an arms race, given a criterion to measure progress, is considered enough for co-evolution to be seen as matching our expectations (from what it does) and what it actually does. This is clearly not the case, as demonstrated in [106], and further shown to be true in [53].

2.3.1 Solution Concepts in Co-evolution

In the quest for understanding co-evolution and designing algorithms in a more methodical manner, the notion of co-evolutionary solution concepts was introduced into co-evolutionary algorithm design research in [53]. As noted in [53], where the main idea is to examine the divergence between expectation and actuality with regard to co-evolution, all the aforementioned problems associated with co-evolution are essentially attributed to a single fundamental problem, that of a lack of rigour in the definition of the co-evolutionary solution concept. All co-evolutionary algorithms optimise according to some solution concept, which may be defined explicitly or implicitly. Ficici [53] has given numerous examples of applications of co-evolution where *"failures to obtain the desiderata*[1] *complexity or optimality indicate a dissonance between the implemented solution concept and that required by the envisaged goal"* [53].

A solution concept is a notion borrowed from game theory. It specifies whether a strategy played by an agent adheres to a set of standards that make the strategy preferable to be used in a problem over other strategies, and thus, retained as a solution to the problem. It separates solution strategies from those that are not preferred or desired as solutions, almost like a filter through which the whole search space of strategies is passed, retaining only the desired strategy or strategies. We do not have the luxury of explicitly scrutinising each strategy in the search space, which leads to sampling strategies (via exploration of the strategy space), evaluating them, and following a gradient in the search space towards the desired solution strategies. The solution concept can thus be specified as using the gradient to separate potentially preferable strategies from those that are not. Co-evolution, which helps specify the solution concept and thus helps solve the search problem of finding the desired solutions, makes finding this gradient harder however, because the evaluation of strategies is relative to the strategies sampled, and not against a static/absolute function that needs optimising. The pressure towards filtering out the desired strategies is built on by channeling the strategies along a certain path in the search space. This path in the search space is however laid out based on an incomplete and changing view

[1] Something that is desired or wanted.

of the space, the current (at any point during the course of the search process) strategies themselves representing the incomplete view of the search space. This is very important in co-evolution because the incomplete view of the search space is also *reactive*, in the sense that it adapts, to the strategies representing the current state of the algorithm, i.e it is *important to find which strategies to interact with so that these strategies (used for the interaction, and in turn, as opponents for evaluation) react or adapt favourably/usefully towards laying out the path.* A secondary search problem (requiring search effort) thus results, that of searching who the strategies must interact with. The opponent strategies that describe the incomplete yet useful view of the search space to adapt with or evaluate against, need attention in their discovery and use.

Thus, co-evolution involves solving a secondary search problems: one that allows for selection of preferable strategies, and the other that allows for the selection of a sample of strategies against which evaluation of any strategy may provide a preferable gradient. *Designing and following the primary and secondary search problem gradients entails a thorough investigation of the elements that make up a co-evolutionary algorithm, and is what we call the* ***implemented co-evolutionary solution concept***[1] *in this thesis.* Note that an algorithm *implements* a solution concept [53]. Two algorithms implementing the same solution concept solve the same search problem, whereas algorithms implementing different solution concepts solve different search problems. If we want to find the subgame perfect equilibrium of a game through co-evolution, then two algorithms which indeed do find it, are said to be implementing the same solution concept of finding the subgame perfect equilibrium. On the contrary, if the algorithms find different solutions, they are solving different problems [53]. Thus, the elements that make up the algorithm need careful design in order for the realisation of the envisaged solution concept from them, while they interact with each other within the co-evolutionary implementation.

This includes the representation that decides the nature of the search space, interaction, evaluation and selection schemes that decide on the nature of the sample and the gradient provided by the sample, and the variation operators

[1]Note that in the Chapters to follow, whenever we use the term solution concept, we mean the implemented co-evolutionary solution concept.

that help traverse the search space. In order to use co-evolution to lead to a certain kind of behaviour one has to decide on these elements for the co-evolving entities, giving us the implemented solution concept. So, given an expected view on what co-evolution should result in, any mismatch in the actual behaviour of the algorithm signifies a mismatch between the desired solution concept (desired strategise to be filtered out) and implemented solution concept.

In [98], solution concepts are interpreted as *superiority criteria*[1] and that the superiority criterion must be well defined in order to necessitate progress. While a sound superiority criterion is indeed necessary to filter out solutions one wants from the search space, when one delves deeper down into the implementation details of the algorithm, this criterion is linked with the elements that co-evolutionary algorithm is made up of, i.e. the kind of interaction that is necessary for evaluation using the criterion, the manner in which individuals are selected given the evaluation criterion, the representation of an individual (defining the search space within which individuals traverse via variation, interaction, evaluation and selection), and, the manner in which the variation schemes work on the representation to help traverse the space. These elements of the co-evolutionary algorithm have to be designed properly in order to make the algorithm address the primary and secondary search problems mentioned in [53]. In fact [98] states that in order to obtain a sound definition of the superiority relation between two individuals in a co-evolutionary algorithm, one needs to define the superiority criterion as well as the opponents against which the superiority criterion is to be used. Looking at this closely tells us that the evaluation of an individual based on the superiority criterion against a set of opponents entails defining the way in which the set will be defined (secondary search problem [53]), the way in which the set of opponents will be used (interaction) for evaluation, and the way in which an individual will carry on via variation, interaction and selection, given the superiority criterion. These operations happen on an individual represented in a certain way dictating its behaviour within the space of possible behaviours, and so, the representation may impinge on the variation, interaction and selection

[1] A superiority criterion is a preference relationship that allows for preferring one solution over the other based on the definition of the solution concept (quality one wants in a solution), given that we do not have the full search space at hand, when pitted against a reference set of opponents.

(as representation defines the search space, which gets traversed using variation, interaction and selection, given the evaluation scheme, and this traversal may necessitate different variation, interaction and selection schemes to be used with different representations in order for a proper traversal of the space), and thus would need proper consideration/definition too, as would the variation, interaction and selection schemes then. The secondary search effort poses the need for the definition of the elements as well. Thus, we consider the entire set of elements that define a co-evolutionary algorithm as defining the solution concept that is implemented.

2.4 Agent Based Computational Economics

Decentralised market economies, as suggested in [145], are complex adaptive systems, made up of numerous agents with adaptive capacities, which engage in multiple parallel local interactions. This local interaction amongst agents results in macro-level regulation (such as shared market protocols and behavioural similarities), which, in turn, feed back onto directing local interactions further. This results in a two way feedback process, local feeding into global and global feeding into local. Though this has been known by economists [71, 108, 132], the machinery to model this two way feedback quantitatively (as said in [145] "*in its full dynamic capacity*") has largely been unavailable.

The machinery available, mostly lead to a top down treatment of the market structure and agent behaviours, with many restrictive assumptions to make an analytic treatment possible. New tools have now become available which allow for modelling complex phenomena associated with decentralised market economies with more realism, for example, learning by trial and error, imperfect competition, endogenous emergence of structure of interaction and open-ended co-evolution of individual behaviours and mechanisms.

A holistic study of economies involves modelling the individual elements that the economy is composed of, made possible with the availability of new modelling tools, and let the system unleash in its full computational capacity. According to [145], this results in the:

> *"... computational study of economies modelled as evolving systems of autonomous interacting agents."*

This is agent based computational economics (ACE). The idea is to see the evolution as happening in a computational laboratory [43, 145], and study the process and outcomes that result under controlled experimental conditions [145]. The point of this thesis is to help ACE further this goal, by having more control on the process often used to simulate the interaction, adaptation and learning mechanisms of social interaction, viz. co-evolutionary algorithms, as we shall see in the Chapters to follow.

The two main objectives of ACE are, for it to help *describe* the process and outcomes as they emerge, and for it to help *design* the process and outcomes of interest, thus having both a descriptive and a normative nature. Moreover, with the advent of the Internet, ACE has grown from being a simulation methodology representing existing or potential economic processes, to being directly applicable in practice (i.e. synthesising real economic processes). The case in point for the latter being, automating economic markets, whereby computational agents either work together with or altogether replace humans (e.g. trading agents on eBay, algorithmic trading in the stock market, market based control [30] etc.).

The essential fields of research within ACE [145], which are directly impacted by this thesis, can be enumerated as follows:

- **Modelling learning:** Representing the learning processes of computational agents is the essential idea here. There have been many computational approaches used for this representation of agent learning, notably, genetic algorithms [27, 39, 79], evolution strategies [58, 59, 149], reinforcement learning algorithms [50, 142], learning classifier systems [93] and any adaptations of learning algorithms specifically channeled towards automated markets [143]. All this work assumes agents as being boundedly rational from the outset, with [143] calling them 'myopically optimal', describing the short sightedness of the processing of information by such agents, as they learn and adapt by trial and error. It was suggested in [57] that techniques from the artificial intelligence should be made use of in modelling boundedly rational learning.

The concern with adopting these algorithms for the purpose of modelling learning, as echoed in [145], is that these algorithms were originally developed for optimality seeking (as opposed to the challenges posed by a *co-adaptive* consideration of these algorithms, i.e. with no objective optimality criterion to optimise against) and not for simulating learning and adaptation in societies. Although, as pointed out in [39], these algorithms, in particular genetic/evolutionary algorithms, when used in a co-adaptive setup (as co-evolutionary algorithms), can provide insights into the workings of learning and adaptation in societies if *interpreted* as models of a learning population, i.e. interpreted as a soico-economic simulation. Optimisation algorithms can be treated as global optimisers with an exogenous (to the agents) fitness function that suggests global properties of interest, e.g. market efficiency, or they can be treated as local optimisers, with an endogenous fitness function seeking the optimal actions for an agent given other agents/environment, which is the case of interest here and where, as [145] notes, the algorithm will need to incorporate some characteristic of actual human decision making behaviour. The idea for the endogenous case is for there to be some predictive power in the simulations.

Other studies on agent learning representations includes [130], where multiple computational trading algorithms were pitted against each other in a double auction tournament and a simple strategy (as opposed to sophisticated learning algorithms) won, suggesting deliberation at a local level of the agent is not always good. Other studies within the realm of auctions are [31, 32, 63].

More pertinently to the ideas promoted in this thesis, that of co-evolutionary solutions concepts, [152] conducts a study using evolutionary algorithms whereby the focus is driven onto the 'level' of learning designed/engineered into the algorithms. The two cases studied are called individual and social learning and are suggested to be modelling the interactions between learning agents, and the way an agent constitute its strategy to play the game, differently. In the case of individual learning, a population of strategies belonging to a firm (which is the agent in this case) only exchanges information within

itself, for instance, by way of evolutionary operators like cross-over (thus, this populations of strategies constitutes the overall game playing strategy of the firm), and in the case of social learning, there is exchange across firms (thus, the population of strategies represents a population of firms, each strategy constituting the overall strategy of a firm). From the artificial intelligence research perspective, the two approaches can essentially be considered as follows. The first considers the use of a ideas like learning classifier system [74] for each agent, thus only exchanging strategies within the agent, whereas the second uses ideas like genetic algorithms where the population contains strategies used by all agents. These two can also be considered as the Michigan [74] and Pittsburg [42] approach to socio-economic simulation respectively, as they are called in evolutionary algorithm design research channeled towards machine learning and optimisation problems. This has lead to many objections in terms of the socio-economic interpretation of evolutionary algorithms, notably [27, 39, 46, 86]. This suggests a need for the consideration of co-evolutionary solution concepts in ACE, for a more systematic study and thus a better understanding of the way the elements that it is composed of, affect simulations. We carry out a study in this thesis in Chapters 3 and 4, laying down an explicit link between co-evolutionary algorithms and socio-economic simulations, specifically when it comes to modelling certain phenomena of interest that may be open to speculation for their translation to evolutionary algorithms overlaying interpretations from the socio-economic learning point of view.

Other questions being looked at in modelling or representing socio-economic learning processes are whether or not the information used by agents evolves during the course of the market processes and in what way this might happen. In some sense, this study focusses on the amount of deliberation that endogenously emerges to be useful for agents and how these deliberations relate to the evolved outcomes [95]. This is a detailed study in bounded rationality, but differs from our work in the sense that, it assumes the use of a genetic algorithm (off-the-shelf, as is common practice). So, it does not show the necessity of a sound interpretation of socio-economic phenomenon like bounded rationality (a case we consider throughout this thesis) in ACE

research when using evolutionary algorithms, and the way a sound interpretation can be made via a methodology, which is the point of this thesis.

- **Modelling market processes:** This line of research deals with understanding market mechanisms by modelling them, e.g. oligopoly [94], and investigating the behaviour of agents modelled in some arbitrary (and sometimes tuned to real data) manner, e.g. genetic algorithms [94], classifier systems [10] etc. The difference from modelling learning (as discussed above) is that this field is concerned with the mechanism per se, and uses learning algorithms off-the-shelf. On the contrary, modelling learning research still uses algorithms off-the-shelf but gives an economic interpretations to them. In both cases, the off-the-shelf treatment of agent learning and adaptation suggests a need for a methodical approach to agent modelling. Evolutionary algorithms have indeed found an application here as well [89]. Trading rules to act within a financial market model are co-evolved and the evolutionary emergence calibrated to macroeconomic data [89]. The work of [58] spans the modelling learning and modelling market processes research areas. On the one hand, it used an evolutionary algorithm for modelling the socio-economic learning and adaptation process of agents, and on the other, it studies these agents in various market settings that involve bargaining over single and multiple issues. In so doing, it studies the conditions under which mechanism/game level outcomes, for example subgame perfect equilibrium in the alternating offers bargaining setting (the game considered throughout this thesis), emerge with evolutionary algorithms doing the learning for agents. Our work spans the two as well, but the main focus is not to study a certain kind of emergence, but instead to propose a methodology that helps co-evolutionary algorithms be used in a more methodical manner, than as an arbitrary choice. The arbitrary choice can lead to mis-interpretations of socio-economic phenomena that may be used to interpret the learning process of agents. We elaborate on a framework which can help enable making interpretations in a more methodical manner in Chapters 3 and 4.

- **Emergence of behavioural norms:** Evolutionary dynamics and bounded rationality are made explicit if one takes a bottom up perspective on agent

interaction and adaptation. Bounded rationality is assumed but indeed an inherent quality of such agents. Over the course of evolution, certain norms on the way agents behave may grow and decay. This is one direction ACE has channeled into. One of the most influential [145] studies in this regard is that of the evolution of mutual cooperation within a society of self-interested non-related agents [12]. The norm here is reciprocity, that allows for cooperative outcomes from agent interactions to emerge. Essentially, there are patterns that develop on the way agents behave as an unintended consequence [132] of local interactions. Further studies along this vein are those considered in [7, 13, 29, 38, 48, 49, 75, 159].

- **Economic networks:** This line of research studies the effect that the agent interaction network has on various aspects of the modelled economy. Moreover, endogenous formation of trade networks has also been looked at [144, 151, 155]. Furthermore, classifier systems have found their application in specific market models, e.g. fish market [85], where buyer loyalty to sellers is questioned. Recent work [47] in the endogenous network formation vein involved studying the emergence (both using an evolutionary game theoretic approach and using co-evolutionary algorithms) of cooperative behaviour in the one shot prisoner's dilemma game. The type of interaction, endogenised by using a reputation mechanism that aids discriminator strategies discriminate between cooperative and non-cooperative opponents, results in cooperative behaviours. Though this is different from the issues considered in this thesis, the fact that co-evolutionary algorithms have found their application here as well, only adds to their wide ranging application, thus needing attention in terms of their proper design.

- **Validation across real and computational agents:** In order for an ACE experiment to tell the causes that lead humans acting within societies to various outcomes, there is a need for the ACE model to be more realistic. Realism of the evolutionary learning process has been questioned often [1, 5, 27, 39, 46, 86, 145]. One line of research in this regard suggests parallel human-subject and computational experiments [4, 5, 6, 9, 25, 99]. This allows for learning processes of computational agents to be guided using

human-subject behaviours. Synergistically, the opposite view of computational agent behaviour providing hypotheses on the root causes of human behaviour, is also a direction [145]. The emergence of money (within a search model of money, as the model is called [145]), is one such study [44]. Reinforcement learning is used as the agent learning process where the agent behaviour rules are guided by human-subject experiments. A further study with computational agent in modified versions of the model suggests outcomes *"roughly similar"* [145] to human experiments. This form of cross-validation is not always possible and raises questions about the validation of computational agents using human-subject experiments, as explained in [117]. This is just one way to validate and a methodical treatment of learning processes within the ACE community is clearly echoed here, as it is in [1, 5, 39, 96, 153]. We propose a methodology in this thesis which links co-evolutionary algorithm design research and ACE, to make the use of co-evolution as a learning process, more methodical in ACE.

Other areas of research [145], which come under the framework of ACE, but are not directly impacted by the work considered in this thesis are:

- **Automated market agents:** Considering reduction of labour costs and search efficiency, computational agents, specific in their optimisation capabilities, have found a use in automated marketplaces to deal with other agents. A study of boundedly rational agents, bounded in the amount of look ahead and self-interestedness employed, is carried out within a negotiation framework in [3]. For such automated scenarios, it should be noted that the line between simulation and reality is blurring. The idea now is to tackle computational agents behaving in automated markets, as opposed understanding economic activity within real, human centric, economic machinery. The impact of automated agents on real markets where both humans and agents interact is one direction gaining ground recently, e.g. algorithmic trading in financial markets [66, 114], and forms a middle ground between studies tailored towards a fully automated or a real market. Simulations tuned to automated agent behaviour, e.g. opponent modelling

[34, 73] is one of the thrust when it comes to designing agents for automated markets. Co-evolution has indeed been used for this purpose [107], but for the purpose of the ACE work considered above, co-evolution still needs a more systematic treatment, as described in the Chapters to follow in this thesis.

- **Modelling organisational structure:** At a more micro level than the study of economic networks lies the study of organisational structure. The idea here is to consider a group, also known as an organisation in economics literature [145], and study the effects of the structure of the organisation on its own resulting behaviour [145]. Both organisational structure and the structure of the network have also been considered together [41], where a specific (to the agent based model) adaptation method is used to adapt the behavioural rules (dictating whether the organisation should innovate or not). This is done to understand how the rules should adapt to structural changes both at the network and the organisational level.

There have also been studies to understand the extent to which learning processes of real world market participants are maladapted to market institutions, giving optimisation algorithms as they are (i.e. without need for matching human like learning processes) a possible application domain [84]. The diametrically opposite question of the extent to which market institutions have evolved or can be designed in order for market participants to be constructed without much regard for them being rational has also been investigated [31, 32, 63]. A combination of the two, from the point of view of informing the design of automated markets, where co-evolution at both the agent behaviour level and the market mechanism level has also been investigated [115, 116].

Viewing games as strategic interaction problems embedded in natural and social processes, agents repeatedly and innovatively tackling these problems over time, evolving the ability to play the games effectively as a result [62, 145], is one of the drive behind modelling the learning and adaptation processes of agents. One of the key issues in ACE dealing with modelling socio-economic learning and validation of simulations, is to effectively gain insight into the dynamics of the simulation for socio-economic games, more complex versions of which may be

intractable [1]. Having gained insights and understood the way desired outcomes can emerge from simulations, can help practitioners with useful insights into solutions for more complex versions of the games, or even generalise to complex versions. The idea here is to ensure predictive content in complex setups of games, given adaptive agents. If the simulation of learning does not lead to desirable outcomes in simple games, it cannot be expected to simulate learning for more complex versions of the games. This is one form of validation, as opposed to validating against human-subject experiments, as carried out with varying degrees of success in [2, 59, 149]. Interpreting the results of the simulations in socio-economic terms would be of little value if the simulation does not do what is required of it, specifically if modelling socio-economic learning. This is the problem considered in this thesis.

Recent work on the use of co-evolutionary algorithms as an agent learning simulation tool [158], again takes the algorithm off-the-shelf. It gives and economic interpretation to the elements of co-evolution and uses the algorithm to meet its needs of optimising strategies for various games under various dynamic exogenous conditions and information asymmetry amongst agents. Our work is different from this at a fundamental level, in that, we suggest not to use evolutionary algorithms in this simplistic, off-the-shelf, manner.

As far as evolutionary algorithms are concerned, the hunch is that mechanising innovation and progression (given innovation) as evolution, helps realise a link or interpretation between socio-economic learning and these algorithms, as alluded to in [65]. This link is not trivial however, and the goal of this thesis is to investigate the link through the lenses of a methodical framework for algorithm adoption (via algorithm analysis and refinements), as we will see in Chapters 3 and 4.

2.5 Bounded Rationality

Simon [138] is attributed to have coined the term bounded rationality. Bounded rationality is a concept within the Social Sciences that takes into consideration the cognitive limitations that humans face, in conformance with which they solve problems or make decisions, that lead to outcomes which are unclear or uncertain.

The limitations that make one unable to gather, retain, transmit and/or process information, and the limitations on the amount of time they have to gather, retain, transmit and/or process information, on route to solving a problem/making a decision, can be seen as limits on rationality, or bounded rationality. Quoting Williamson [157] citing Simon, *"boundedly rational agents experience limits in formulating and solving complex problems and in processing (receiving, storing, retrieving, transmitting) information"*. For example, a decision maker may not have the resources to find the optimal solution to a problem at hand, which leads to the decision maker making assumptions about the problem in order to arrive at a solution, thus greatly simplifying the problem/possible choices of solutions to the problem. In computational terms, this limitation can translate into an agent being incapable of gathering, retaining, transmitting and/or processing the information or having limited time to gather, retain, transmit and/or process the information that ideally can lead an agent to behaving as if it possessed complete information and unlimited processing capabilities. Fully rational behaviour is only possible if decisions can be made such that they lead to outcomes that are clear/certain during the time of processing making such decisions, and that these outcomes do not change whilst processing.

As such, an agent tries to *satisfice* [137] rather than optimise its payoff (from a world within which it is situated), i.e. the agent aims for a satisfactory rather than an optimal solution, subject to the information it has been able to gather, retain, transmit and/or process, and the time it has to do so. In other words, satisficing is the idea that, an agent using a strategy to play a game would attempt meeting some criteria for the payoffs from the strategy to be satisfactory enough or adequate, rather than optimal. The agents thus look for an outcome from the game that satisfice their needs, where these needs may, for example, be described as utility functions in game theoretic and computational approaches to solving games. Simon [138] further suggests that agents make decisions using heuristics, for example, experience based decisions or common sense based decisions, so that they may be able to manage the deliberative costs that may incur if they were to base their decisions on exhaustive reasoning. Note that this notion of exhaustive reasoning, when seen in the computational realm of solving games using agents, may be seen as a computational agent exhaustively

searching for a solution, which indeed is what ACE and indeed computational approaches to solving games relieves us from doing. In agent based simulations, where agents have limited computational capabilities, such agents are deemed boundedly rational from the outset [131, 145].

One of the ways bounded rationality has been modelled in socio-economic literature is via the notion of a *"trembling hand"* [133]. The idea here is that a player would not always use its strategy such that the moves that it makes in accordance with its strategy are exactly what the strategy specifies. The players thus make moves with a *trembling hand*. The way this is modelled is by there being a probability associated with the players, according to which they make moves erroneously. This is one notion of bounded rationality that we explore in the thesis in Chapter 4, wherein we specify it in concrete terms. There have been other specification of bounded rationality in socio-economic literature [60, 129], and these could also be investigated in the context of ACE, as we utilise the *trembling hand* specification, in accordance with the methodology presented in the thesis, but it is not necessary for us to do so for the purposes of the thesis. However, we encourage the interested reader to investigate different specifications of the term as we investigate the notion of *trembling hand* in the thesis.

As far as ACE is concerned, the *trembling hand* notion of bounded rationality has indeed been used within simulations recently, for example the work in [79, 122]. The work in [79] considers representing agent strategies as finite state machines, and uses genetic algorithms to understand the kind of machines that may result over the course of evolution, having evolved with the specific bounds on the way they function. The essential idea is that an agents play the iterated version of the prisoner's dilemma [55] game and make moves, with a chance of these moves being opposite of what the agent intended, or the opponent's moves are reported to be the opposite of what the opponent actually chose, the moves being to either *cooperate* or *defect* in a given round of the iterated game. The machines that result over the course of evolution are shown to behave less cooperatively and thus bounded rationality is shown to hamper the evolution of cooperation. This notion is also used in [122] to define bounded rationality, where it is shown how bounded rationality reduces the number of equilibrium outcomes in infinitely repeated normal form games. Both cases however, are not concerned

with the main theme explored in this thesis, that of adopting co-evolutionary algorithms for socio-economic simulations methodically. Most of ACE indeed assumes for bounded rationality to be present in the simulations *implicitly* [131].

2.5.1 Implicit Assumption of Bounded Rationality in ACE

Although ACE relieves us from using the game theoretic assumption of full rationality within players or agents, it however provides too much freedom in terms of what bounded rationality may mean within simulations, i.e. it does not make any explicit assumptions about the rationality of agents. Bounded rationality is seen as implicit in the design choices made about the algorithm, which is then used to simulate socio-economic learning [131]. Quoting Safarzynska and van den Bergh [131], *"Bounded rationality is implicit in many evolutionary economics models or results from specific choices made with regard to the other model components (e.g., selection models)"*. They also state that *"a variety of assumptions regarding boundedly rational behaviour can be encountered in evolutionary-economic models. In many cases, they are introduced ad hoc without clear empirical, experimental or theoretical support"*. They suggest that theories in behavioural economics research (for example, theories in [16, 61, 67, 82]) can help with laying a foundation for specifying bounded rationality in these models.

The fact that simulations are assumed to incorporate bounded rationality, makes it very hard to understand what role it plays within these simulations and indeed, can give rise to bounded rationality being mis-interpreted as a term, a case we consider throughout the thesis, in order to elaborate on a methodical adoption of co-evolutionary algorithms for socio-economic learning simulations. Coupled with the off-the-shelf adoption of co-evolutionary algorithms for socio-economic simulations, the abuse of the term is not entirely impossible, as we do show in Chapter 3. Although we do not profess that the assumption of bounded rationality within socio-economic simulations, as done in ACE, to be wrong, we do want ACE research to pay head to scrutinising the assumption in this thesis, before any conclusions are drawn from the simulations that may in fact rely on the role of bounded rationality within these simulations. Simply put, we emphasise that if we do not know what bounded rationality means within simulations, we

cannot reason about the simulations in terms of bounded rationality.

2.6 Disciplining Efforts for Co-evolution in ACE

Brenner [22] says that, *"learning models are indeed usually chosen without much justification in the literature"* dealing with modelling socio-economic learning within ACE, and that *"the use of learning models is often criticised due to the lack of a common model and the ad-hoc choice of specific models"* [22], commenting on the lack of a methodology for making this choice. Talking about the choice of an algorithm, made by practitioners of ACE, from the choice of algorithms one has within artificial intelligence (not limited to co-evolutionary algorithms alone that is) for socio-economic simulations, Brenner [22] further says:

> *"...there is a strand of artificial intelligence and machine learning. In general, a tendency has been observed in recent years to borrow methods from other disciplines. These models have become increasingly complex in recent years, mixing such features as evolutionary algorithms, classifier systems, fuzzy logic and neural networks. It is not always clear what the researchers in this field aim to do. Some seem simply to believe that their learning models describe real learning without looking at any evidence for this. Others seem to aim at creating learning models that perform well or can solve problems that in nature only humans can solve. Finally, there is those who claim a correspondence to reality."*

The above concern is indeed also present when simply considering making a choice of or using a particular co-evolutionary algorithm for socio-economic learning simulations. The algorithms are adopted or used in an *unconsidered* [145] (not rigourous) fashion. Despite the widespread use of learning and adaptation algorithms in ACE, very few researchers have looked at the off-the-shelf nature of this usage. The need for disciplining learning and adaptation algorithms channeled towards modelling socio-economic learning is clearly echoed in [145]. This thesis essentially deals with this very issue, but specifically for the use of co-evolutionary algorithms in ACE research for modelling socio-economic learning, and shows one

way this usage can be made more methodical. Following is a look at the previous efforts that have been made in this direction.

Often, with the use of evolutionary algorithms for socio-economic learning, the parameters of these algorithms are directly derived from the socio-economic model parameters [2]. According to [2], this direct mapping has been shown to be the cause for particular kinds of evolutionary algorithms (the ones often used for simulations, and taken for granted for their applicability), not doing what they were intended to do, e.g. learn to achieve a certain kind of learnt/adapted/emerged outcome. It has thus been stressed that algorithmic and model parameters be treated separately [2]. The way this separation is proposed to be looked at is by comparing the two *widely used* kinds of co-evolutionary algorithms (individual and social learning approaches) for socio-economic simulation. This comparison is done with respect to their convergence properties and robustness to evolutionary algorithm parameter value variations [2].

Evolutionary algorithms, having originally been designed for optimisation problems, raise questions on the tuning of their parameters when used for socio-economic simulation. A goal in the above mentioned previous work [2] is to reestablish guidelines for tuning evolutionary algorithms for socio-economic simulation. The reason is for the simulations to be robust [2], rather than sensitive to changes in problem instances.

Note that the two approaches of individual and social learning are taken as a given, and these are then compared for their learning capabilities in the Cournot oligopoly game [35]. In the individual learning approach [39, 152], one or a set of strategies (e.g. a population of evolving strategies) represents one agent alone, which does not exchange its traits by way of evolutionary operators like cross-over, with other agents, but instead updates its own set of strategies using such operators, though the agents indeed interact. In social learning [11, 28, 121], the evolutionary algorithm population as a whole represents strategies shared amongst agents, i.e. strategies that they may choose from for their operation in the socio-economic situation considered. Note that these two approaches differ in what a population of evolving strategies means in socio-economic terms, leading to changes in the way the individual agents interact, are evaluated etc. This changes the co-evolutionary solution concept being implemented, as discussed in

Section 2.3. It is shown that the first approach leads to premature convergence and lower profits for agents in the Cournot game. This demonstrates the need for the notion behind co-evolutionary solution concepts to be considered in ACE, apart from the parameter settings of the particular concept in use, the latter being the main cause for concern in [2].

Our take in this thesis is that, co-evolutionary solution concepts can also be parametrised from the point of view of socio-economic phenomenon of interest which may have lead to anomalies in the outcomes expected from the co-evolutionary process, and may have been mis-interpreted if not looked at carefully. It is clearly mentioned in [2], that a fundamental methodology for designing co-evolutionary simulations needs attention and is one alternative to understanding and improving the quality of simulations for socio-economic problems. We indeed take this view in this thesis. Whereas [2] considers evolutionary algorithm parameters that do not have a direct economic interpretation, and studies two co-evolutionary solution concepts for their robustness in outcomes (results being similar or not), when the parameter values are varied, we focus on a methodology for the careful design of co-evolutionary solution concepts (Chapter 3), parametrising them from the socio-economic point of view (Chapter 4). This enables understanding the relationship between the socio-economic phenomenon of interest and the simulation methodically, which may further guide moulding the algorithms for soico-economic simulation, if needed.

Of major influence in terms of disciplining the use of evolutionary algorithms for ACE is [39], which revolves around understanding the ways in which, as [145] puts it:

> *"... particular aspects of the implementation strongly influences the set of potential long run outcomes."*

The caution one should use in using genetic algorithms for socio-economic simulation is much appreciated in [39]. The question of whether these algorithms are just a technique (and as such other algorithms may be equally worth considering for socio-ecnomic learning), or whether they can indeed model learning behaviour within societies is raised [39].

There is some work that suggests evolutionary algorithms doing better at simulating socio-economic learning than other techniques. Evidently, simulations using evolutionary algorithms yield better approximations to empirical results than least square learning for instance [5, 39]. There is evidence that *some* experiments with evolutionary algorithms for building trading strategies in auctions resembles observed human strategies [4]. Moreover, it is stressed in [39], that the fact that evolutionary algorithms (more specifically, genetic algorithms) are used to simulate socio-economic learning is that there is no rigourous empirical foundation of how humans form expectations over outcomes from socio-economic situations and their own behaviours. Quoting Simon [139]:

> *"Armchair speculation about expectations, rational or other, is not a satisfactory substitute for factual knowledge as to how human beings go about anticipating the future, what factors they take into account, and how these factors, rather than others, come within their range of attention."*

Since there is no rigourous study in economics and psychology that allows us to specifically model socio-economic learning, the best one can do is to *analyse* models which "*describe some plausible features of actual learning behaviour*" [39] and match these models with game theoretic and experimental economics studies. Doing this is suggested to allow for judging their suitability as socio-economic simulators. At the least, in so doing, one may gain insights into the dynamics of actual adaptive behaviour, even though the process of picking a learning model may be ad-hoc [39]. Classifier systems (modelling individual learning) have successfully been used to find equilibrium in a model of money [93], not only for the model for which the equilibrium was known (analytically), but also finding the equilibrium in a model where it was not. This suggests that techniques from computational intelligence may not only be used to simulate socio-economic learning but also to compute equilibria in complex games where analytic computation becomes cumbersome [93]. Using the social learning approach, the equilibrium was again found in the simpler model [91].

The interpretation of evolutionary operators from the socio-economic point of view is called into question [39] and deemed unsatisfactory [39]. Parameter

values of these algorithms is another questionable issue, where a direct mapping between the values and socio-economic values (if interpreted that way), is not known. Different parameter settings result in different results from evolution, so this is important [39]. This issue has been addressed in [2] to some extent, as we explained earlier on in this Section. It has also been addressed from the angle of calibrating parameters using experimental data [44]. Different strategy encodings or representations lead to different outcomes too [153]. So, a link between representations and the economic interpretation of representations is another issue needing attention [39].

The main concern expressed in [39] is that there is *no mathematical theory* which may explain the results of simulations using genetic algorithms. This makes it impossible to predict the behaviour of genetic algorithms without actual simulations. Doing this can guide the design of genetic algorithms for socio-economic simulation when particular kinds of outcomes (e.g. equilibrium or deviations thereof as stable states) are desired. Analysis of genetic algorithms is carried out from the point of view of Markov chains [105], extended to include characteristics that differentiate co-evolution from evolution for optimisation, i.e. the fitness of an agent changes as the opponents change, and the notion of optimality changes to equilibrium finding. This leads to theoretical results that suggest what the genetic algorithm might evolve. For some games this kind of analysis is possible and for other it is not [39]. In the former case, the genetic algorithm simulation shows a match with theoretical outcomes. In the latter case, the genetic algorithm is able to learn cyclic equilibria. This is interesting from the point of view of understanding what a particular kind of evolutionary algorithm (though taken off-the-shelf), may result in when used as a simulation. What we suggest in this thesis however, is a way of making the choice of an evolutionary algorithm easier (methodical) from an empirical point of view. Our work in this thesis complements the work in [39] such that, instead of treating an off-the-shelf evolutionary algorithm theoretically, one can envisage using our methodology to help narrow down on an algorithm first, and then conduct a theoretical analysis for further confidence.

Unaware of the existence of the work in co-evolutionary algorithm design, more specifically co-evolutionary solution concepts, a study is undertaken in [8],

wherein the differences between individual and social learning are examined from the point of view of convergence to the equilibrium in a Cournot oligopoly game [35]. The main question, essentially for the individual learning case, concerns the changes in the genetic algorithm learning model [5] needed so that it can be engineered towards selecting a particular type of equilibrium, which was otherwise not possible using genetic algorithms, but was with classifier systems [152] representing individual learning. The authors conclude that the manner in which fitness was being calculated and assigned to individuals in the population representing the agent, and the value of a certain parameter, resulted in the genetic algorithm not converging to the equilibrium. They show this by systematically changing the genetic algorithm setup, guided by the classifier system setup. Though unaware of their existence, this work essentially echos the need for viewing co-evolutionary algorithms through the lenses of co-evolutionary solution concepts. This is evidence that ACE research has grown but somewhat independently from co-evolutionary algorithm design research. The aim of our work is to bridge this gap. In some sense, they modify the co-evolutionary solution concept, working on the fitness evaluation systematically, giving them the desired behaviour from the algorithm using the conflicting results obtained using another solution concept. Our work (in Chapter 3) looks at the elements of the co-evolutionary solution concept and obtains desired behaviour without the need for a separate solution concept to guide the changes needed. This suggests that co-evolutionary solution concepts, if explicitly examined, are a powerful way to engineer the algorithm, and we stress this point throughout the thesis.

2.7 What is Missing?

An important point to be kept in mind is that co-evolution is a search process, which gets interpreted as simulating socio-economic learning in ACE. Problems in co-evolutionary algorithm design research, are hence not too far away from problems in using co-evolution as a socio-economic simulator. ACE, as a field of research has often used socio-economic terms to understand and explain the implementation details of co-evolution, with little rigour. Thus, a socio-economic interpretation to the process and the results that come about from it, is a subject

of much criticism [39]. As a general case for instance, selection, crossover and mutation are interpreted as 'imitation of the successful' [39], information exchange [1], and innovation or making mistakes [1], respectively. Moreover, bounded rationality is generally assumed to be implicit in these algorithms [131], *without scrutiny*, i.e. without considering understanding the real role it plays within them. These are forced (i.e. not thoroughly investigated) qualitative meanings associated with the co-evolutionary algorithm and suggest a passive treatment of the problem of modelling socio-economic learning. The simulations thus get mis-interpreted in socio-economic terms, or indeed, the socio-economic terms used for explaining the co-evolutionary simulations get mis-interpreted within these simulations. This thesis considers this practice inappropriate and explicates on a *methodical framework* for analysis and systematic refinements to co-evolutionary algorithms, for them to simulate socio-economic learning, specifically so that mis-interpretations of socio-economic terms interpreting these simulations be avoided.

In order to match simulation and socio-economic phenomena, i.e. to avoid mis-interpretations, refinements to the algorithms may indeed be necessary. There is a dissonance between interpreting co-evolution as a socio-economic simulator and co-evolution inherently being an adaptive search process. A conformance between simulation and search is thus in order, otherwise simulating socio-economic interaction in order that the agents adapt towards reaching a certain type of behaviour (rummaging through the search space of behaviours) would have little meaning, i.e. simulations will not do what they are meant to.

Co-evolutionary solution concepts are thus explicitly dealt with for making the refinements necessary, in order to show how to make simulations achieve what they are meant to. This allows for the avoidance of mis-interpreting socio-economic phenomena that, erstwhile, may have been mis-interpreted within simulations, by allowing us to challenge and scrutinise the meaning of the socio-economic phenomena within simulations through the lenses of co-evolutionary solution concepts. This results in a top-down (from the point of view of there being a top-down guiding notion of desired outcomes from the simulation) mis-interpretation avoidance methodology, which is the *first integral part of a methodical framework* that we lay out and apply (for the case of bounded rationality as the phenomenon of interest) in this thesis in Chapter 3. Co-evolutionary solution

concepts are thus promoted to be made use of within ACE research.

Having challenged and scrutinised the phenomena, these phenomena may still not have obvious links with the co-evolutionary algorithms considered for simulations, and thus may still get mis-interpreted within simulations. This is because we may only have uncovered how the phenomena do not affect the simulations on the application of the first integral part of the methodical framework. A methodology for using the idea behind co-evolutionary solution concepts together with an explicit expression of such socio-economic phenomena of interest (which we call reconciliation variables), is thus proposed and applied (for the case of bounded rationality as the phenomenon of interest) in this thesis as well. This becomes the *second integral part of the methodical framework* we lay out and apply in the thesis in Chapter 4. This is done by associating the phenomena with the solution concept, enabling their parametrisation and quantification. Thus, a systematic bottom-up (from the point of view of the socio-economic phenomena) co-evolutionary solution concept refinement methodology results, which leads to a better understanding of the phenomena and its relationship with the co-evolutionary algorithm, thus allowing for the avoidance of mis-interpreting the phenomena altogether from within the simulations.

In essence, a marriage between co-evolutionary algorithm design research and ACE, when adopting or using co-evolutionary algorithms for simulating socio-economic learning, is thus conducted in this thesis, with the idea that mis-interpretations of the algorithms in socio-economic terms (and vice versa, i.e. mis-interpretations of socio-economic phenomena within simulations) be explicitly and methodically avoided. As such, the methodology (a methodical framework for analysing and refining co-evolutionary algorithms) we present in the thesis should help increase the value of the adoption or usage of co-evolutionary algorithms, when used as simulations of socio-economic learning by ACE practitioners.

Chapter 3

Engineering Co-evolution

3.1 Introduction

Co-evolution is the process of adaptation, that occurs amongst agents interacting with each other in some manner, in turn revealing a reward structure, towards increasingly complex or refined behaviours. It has been used for both the evolution of game playing strategies [80] in order to achieve a certain type of behaviour (amongst other applications viz. sampling test cases for fitness evaluation to improve search efficiency [72], open-ended emergence of behaviours [140], reduction of domain knowledge required to learn complex control [90] etc.) and at the same time using games as a test bed to understand the co-evolutionary process properly [28, 36].

However, the reported successes of co-evolutionary applications are counter balanced by many failures. These failures are often attributed to co-evolutionary properties such as *cyclic dynamic* [106], *mediocre stable-state* [118], *collusion* [118], *forgetting* [53], *focussing* [154] etc. It has been shown that often these pathologies imply a lack of rigour in the definition of the *solution concept* used for the co-evolutionary process [53]. All co-evolutionary algorithms optimise according to some solution concept, which may be defined explicitly or implicitly. Ficici [53] has given numerous examples of applications of co-evolution where *"failures to obtain the desiderata complexity or optimality indicate a dissonance between the implemented solution concept and that required by the envisaged goal"*

[53].

Game theory provides a powerful framework to understand interactions between entities. These entities could be players with complete knowledge of their environment, players with incomplete knowledge of the environment, and more pertinently, computational agents interacting with each other under various assumptions about their deliberative abilities. Theoretical results from game theory e.g. Subgame Perfect Equilibrium (SPE) [133] as it is called in a sequential game like bargaining, give us a useful idealised result in terms of what outcomes one may expect with players interacting under theoretical assumptions of full rationality and complete knowledge. Although these theoretical assumption are far too strict and do not in any way model the reality of the particular interaction or game under question, they are indeed required for a closed form solution to the game. The value of these theoretical outcomes is arguable, at least in economics and psychology research. However, for the purpose of understanding and disciplining bottom-up approaches like co-evolution, they are indeed a boon. These solutions give us what is known as a 'solution concept' or a 'goal' to aim for, without which evaluating the outcomes from a co-evolutionary process becomes extremely hard, or even impossible, for the very fact that co-evolution can have an open ended dynamic. To put a handle on the process, one needs a definition of the quality of the solutions one expects from the process. Game theoretic solutions give us these quality attributes which we may use in order to engineer (build, validate, and refine) co-evolution towards achieving them in the solutions that emerge out of the process. Note that, whether or not game theoretic outcomes must be used as a goal for co-evolution is an entirely different matter. The goals could well be defined as deviations from game theoretic outcomes, as we will consider in Chapter 4. One could as well use experimental data from human experiments for example, if this data were indeed available, to construct the goal. The idea is to have a pre-defined goal to aim for which is specific enough for validating the simulation outcomes. We are interested in engineering the process of co-evolution to achieve our pre-defined goals.

Apart from game theory helping validate co-evolution, there is a second advantage and indeed, our main purpose, in carrying out this activity. Agent based computational economics (ACE), as a research field, has seen a use for

co-evolution as a mechanism to model socio-economic learning and adaptation in order to see what emerges (as a culture dish approach to economics), understand how the emergence came about, and give socio-economic interpretations to the co-evolutionary process and the results that emerge. Socio-economic learning has further been used in this manner to simulate equilibrium selection. One of the reasons this is interesting is that this gives support to the designed co-evolutionary algorithm to be applicable for equilibrium selection in game settings where the theoretical equilibrium is intractable. Another reason is that this can lead to 'what-if' scenarios that may inform a practitioner about details that they may miss when trying to find a closed form solution for an economic game.However, an important point to be kept in mind is that co-evolution is a search process, which has been interpreted as simulating socio-economic learning within ACE. Problems discovered in co-evolutionary algorithm design research, are hence not too far away from problems in using co-evolution as a socio-economic learning simulator. ACE, specifically when simulating socio-economic learning, due to the field growing largely independently of co-evolutionary algorithm design research, has often used ideas and notions from economics to understand the implementation details of co-evolution, with little rigour. Thus, an economic interpretation to the process and the results that come about from it, is a subject of much criticism [39, chap 2.], as these may in fact be mis-interpretations. As such, the socio-economic terms or phenomena used to interpret the process and results get mis-interpreted within simulations. There is a dissonance between interpreting co-evolution as a socio-economic learning simulator and co-evolution inherently being an adaptive search process. A conformance between simulation and search is thus in order, otherwise simulating socio-economic interaction and learning in order for agents to adapt to reach a certain type of behaviour (rummaging through the search space of behaviours) would have little meaning i.e. simulations will not do what they are meant to.

This Chapter first (Section 3.2) lays out the first integral part of our methodical framework, for the case of avoiding mis-interpretations of socio-economic phenomena of interest within simulations from the top-up, using the notion behind co-evolutionary solution concepts. The Chapter then takes us through a discussion on how co-evolution has been used within ACE (Section 3.3), how

it has often been mis-interpreted, leading to misleading conclusions about what it can achieve as a mechanism for socio-economic learning (Section 3.4). As a case study, we focus on applying our methodology, thus analysing and refining the co-evolutionary solution concept, by considering the phenomenon of bounded rationality. Bounded rationality has been mis-interpreted as being the reason for co-evolution to fail to achieve the desired results previously [59, 149] (Section 3.5). The notion of bounded rationality hence gives us a point of focus to explore and understand, together with understanding whether or not it hinders co-evolution from being a useful socio-economic learning mechanism such that co-evolution achieves the desired results. Using the notion behind co-evolutionary solution concepts and applying our methodology (Section 3.6), we show how and that indeed we can, achieve the desired results, which in our case is equilibrium selection, with boundedly rational agents. We thus reveal the misuse of co-evolutionary algorithms by way of mis-interpreting bounded rationality, which lead to mis-interpretations of the process and results thereof, and show how to avoid doing so using our methodology. Section 3.7 comments on how our methodology generalises to more complex versions of the games considered, and Section 3.8 comments on the issue we continue investigating in the next Chapter.

3.2 Framework Part 1: Avoiding Mis- interpretations From the Top-down

3.2.1 Description of the Framework

Given that one has a co-evolutionary algorithm for simulating socio-economic learning and adaptation at hand, we want to analyse and refine the algorithm so that the simulation may not get mis-interpreted in socio-economic terms. Recall from our objectives (see Chapter 1) that we want to handle such mis-interpretation of simulations by focussing on the socio-economic phenomena of interest that is being interpreted with little scrutiny within simulations, and thus being mis-interpreted. In essence, we want to show how to analyse and refine co-evolutionary simulations such that socio-economic phenomena of interest may

not get mis-interpreted within them. Here, we lay out the first part of the framework that we want the ACE community to follow in order to achieve this. Note that the first part of the framework essentially allows for the avoidance of mis-interpretations of socio-economic phenomena of interest within simulations from the top-down, using the notion behind co-evolutionary solution concepts. The following guidelines lay out the first part of the framework:

1. Ask the question, what the solution concept/goal/desired outcome is, that we want the co-evolutionary algorithm to have agents converge towards? This could be, for example, convergence to the equilibrium, or some other desired outcome of choice.

2. Reduce the population size in the algorithm to get a handle on the co-evolutionary dynamic, if necessary and possible. The necessity becomes evident if the dynamic cannot be explicitly understood with a population. This allows for following the co-evolutionary dynamic and indeed understanding it in explicit algorithmic terms.

3. With the goal in mind, work on the elements that specify the algorithm, to understand how the elements effect the goal and co-evolutionary dynamic. This analysis of the elements, helps refine them in order to achieve the goal under question. Note that working on the evaluation, interaction and selection schemes may need greater attention than the variation scheme and representation used in the algorithm, because the selection pressure largely depends on these three. Changing the representation however, can necessitate a change in the other elements, for the desired goal to be realised.

The socio-economic phenomenon of interest that we consider scrutinising in this thesis is bounded rationality. As a case study within this Chapter, we utilise the above mentioned framework to show how mis-interpretations of bounded rationality can be avoided from the top-down from within previous work [59]. Chapter 4 lays out and applies the second part of the framework.

3.3 Co-evolution in ACE (a modelling and predictive tool)

3.3.1 Modelling Socio-economic Learning and Adaptation

Agent based computational economics (ACE), as a research field, has been using co-evolution as a way to model social interaction and learning, in order to understand the process by which certain socio-economic outcomes emerge, intending to ensure predictive content through mimicry. Agents interact locally, which gives rise to macro phenomenon (macroeconomic in this case), which feed back into affecting local interactions further. This two way feedback [145] has rendered co-evolution as a potentially viable methodology (co-evolution has a similar thematic structure when it comes to this two way feedback) to model it, with the hope that both descriptive and normative concerns expressed by practitioners will be attended to. Descriptive being the need for a constructive explanation of global emergent behaviour, and normative being the understanding of peculiarities of entities that form part of the model [145], e.g. behavioural norms that the agents may evolve towards. Modelling the learning process of computational agents has been one of the main areas of research as far as ACE is concerned. A broad range of algorithms have been used to represent this learning process and as such become candidates for modelling socio-economic learning. The list includes evolutionary algorithms [27, 39], reinforcement learning algorithms [50, 142], classifier systems [93] etc. Given that a lot of these algorithms were originally employed for optimality seeking (as opposed to the challenges posed by a co-evolutionary or co-adaptive consideration of these algorithms), it is known that caution must be used if they are to be applied for socio-economic modelling [145]. Viewing games as strategic interaction problems embedded in natural and socio-economic processes, agents repeatedly and innovatively tackling these problems over time, evolving the ability to play the games effectively (though effective play is not the same as optimising a static function) as a result [62, 145], is the drive behind using the above methods. As far as evolutionary algorithms are concerned, mechanising innovation and progression (given innovation) as evolution, helps realise a link between socio-economic learning and these algorithms, as alluded to in [65].

The motivation for using these algorithms thus has roots in the intuitive notion of optimisation as progression of the agents towards refined behaviours (arms race for example, much the same as the intuition behind early co-evolutionary algorithm design research), which is not correct when one adds a co-adaptive component into the picture, as we know from co-evolutionary algorithm design research. The use of such algorithms with a co-adaptive component, which is what is needed for socio-economic simulations, has however remained ad-hoc. It is now common within the field of ACE to consider one of the two class of algorithms (also known as levels of learning) for the purpose of socio-economic learning simulations, these classes being *individual learning* and *social learning*. These two classes stem from the initial work carried out in the field of evolutionary computation, more specifically evolutionary learning, whereby two approaches (or schools of thought) known as the *Michigan* [74] and *Pittsburg* [42] approach to evolutionary learning were made prominent. These approaches differ in the way a solution to the problem is constituted from an evolving population of solutions. In the Michigan approach, the population as a whole constitutes a solution (individual being partial solutions), whereas in the Pittsburg approach, each individual in the evolving population constitutes a solution to the problem. Considering one agent (or the strategy used by the agent to play a game) as being this solution discussed above, these two approaches directly map to the individual learning and social learning classes commonly considered to make a choice of an algorithm within ACE. Choosing an algorithm residing in one of these two classes, even because of a class being more popular amongst practitioners [152], rather than due to a rigorous study backing the usage, is not uncommon.

There are various reasons why modelling socio-economic learning using evolutionary algorithms may be considered interesting. One deals with effectively gaining insight into the dynamics of the simulation for games, more complex versions of which may be intractable, thus helping practitioners with useful insights into solutions for these complex versions. Another deals with using these algorithms as models of human learning, where efforts to calibrate the algorithms to empirical decision making data are carried out as well [94]. There have also been studies to understand the extent to which learning processes of real world market participants are maladapted to market institutions, giving optimisation

algorithms a possible application domain [84].

Making a methodical choice of the algorithm that may represent socio-economic learning, and thus be a good candidate for carrying out such simulations, is a line of research still in its infancy. Researchers have expressed co-evolutionary algorithms to exhibit promise for modelling socio-economic learning, and at the same time concerns about their ad-hoc usage [39, 145]. Indeed, very pertinently to the focus of this thesis, influences on the outcomes of simulations given simulations differing in the level of learning mentioned above (bearing a resemblance to the idea of a co-evolutionary solution concept, further discussion of which is carried out in Section 3.6) has been touched upon [152], amongst other recent efforts, which we discussed in Chapter 2 (Section 2.6).

3.3.2 How Viable are EAs as a Tool for Socio-economic Simulations?

In general, interpreting the process of evolutionary algorithms and the parameters and intricacies of these methods, as learning with socio-economically sound entities within the context of a socio-economic game, specifically when considering the co-adaptive component within these algorithms (i.e. considering co-evolutionary algorithms), poses problems. The choice of the evolutionary algorithm used for simulating a socio-economic situation in question has in general been ad-hoc. A very simple way to look at this problem is that since evolutionary algorithms do not model natural selection in its original and complex form, if we consider socio-economic phenomena as an offshoot of the natural evolutionary process (indeed, co-evolutionary process), it is not hard to see that at least an ad-hoc choice of an algorithm will not do justice to the idea of simulating the phenomena. In fact, given the issues that have been discovered that hinder co-evolutionary algorithms from doing what is expected of them, some of which we discussed in Chapter 2 (Section 2.3), an ad-hoc choice of these algorithms for the purpose of explaining or indeed growing socio-economic processes, does not sound reasonable. This can lead to the co-evolutionary process and the outcomes there of, being mis-interpreted in socio-economic terms. In order for this mis-interpretation to not happen (i.e. to match simulation and socio-economic phenomena), refinements

to the algorithms may indeed be necessary. To answer whether co-evolutionary algorithms are indeed a viable tool for socio-economic simulations, we would have to consider systematic refinements to off-the-shelf algorithms, and see if we can get rid of such mis-interpretations. The method of design of these algorithms for the purposes of simulations, thus needs a fundamental change.

3.4 (Mis)Interpretation of Co-evolutionary Simulations

3.4.1 (Mis)Interpretation of the Process and Outcomes

Based on some empirical evidence that evolutionary algorithms may lead to desired or close to desired phenomena, it is suggested in [39] that some of the features of these algorithms represent the effects appearing in a learning population well. Along the same lines, for some simple bargaining games, it is noted in [59, 149], that evolutionary algorithms do indeed come close to modelling the desired phenomena, but they fail to do so for more complex games. An evolutionary algorithm is interpreted in terms of socio-economic phenomena such that, the elements that define the algorithm are given a socio-economic meaning. As a general case for instance, selection, crossover and mutation are interpreted as 'imitation of the successful' [39], information exchange [1], and innovation or making mistakes [1], respectively. Moreover, bounded rationality is generally assumed to be implicit in these algorithms [131], *without scrutiny*. These are forced (i.e. not thoroughly investigated) qualitative meanings associated with the evolutionary algorithm and suggest a passive treatment of modelling socio-economic learning, and indeed an off-the-shelf treatment of the evolutionary algorithm considered. One can only say that if the process is interpreted in such a manner, the outcomes may not always match up to what is desired of the evolutionary algorithm. It is in fact shown in [39] that particular aspects of the implementation of the co-evolutionary algorithm can strongly influence the outcomes from the process.

As a concrete example of this non-rigourous treatment of the subject, a case to point out is that agents, in such models, are assumed to have little apriori

knowledge about the game and the opponents, and it is by trial and error that they gradually adapt towards desired/optimal solutions. This deems them to be boundedly rational from the outset [59], as also suggested in [1]. This is general practice within ACE [131]. Still, interpreting outcomes when they deviate from the desired outcomes (not matching rational equilibrium outcomes for example), as being caused by bounded rationality (for example, as presented in [59]), suggests *assuming the methods would never work* for what is required of them. Actively understanding the cause for deviations may however give further insight into whether this is indeed the case, and if not, what bounded rationality may mean in the context of the simulations considered. An investigation of this kind may benefit ACE concretely, as opposed to the field being mislead by way of unscrutinised assumptions. For example, for the case of bounded rationality, if the assumption is proven to be correct, we will learn what bounded rationality means within simulations at the algorithmic level, i.e. the algorithmic details that cause the deviations. But, and very importantly, if the assumption indeed turn out to be incorrect, we will learn what bounded rationality does not mean within simulations, thus avoiding mis-interpreting it within simulations. In both cases, we can then further consider how it may explicitly and tangibly be varied within simulations (something we consider in the next Chapter), because having a concrete handle on it necessarily requires for it to not be mis-interpreted within simulations.

3.4.2 Forcing Interpretations vs. Engineering Co-evolution

If the match between what is required of the outcomes and what actually emerges, is not close (to some degree), it is unclear as to why computational methods should be used at all, other than providing 'what-if' scenarios. In our opinion though, what-if scenarios are simply too hard to evaluate for their usefulness other than them being just a possible result out of the myriads of results, many of which may in fact be more useful than the one that emerges. It has been noticed that interpreting simulation results, where there is indeed a mismatch, has largely been 'hand-wavy' and so, not easy to believe for its soundness.

The fact that co-evolution has been used for agent based simulations, one

would expect there to be efforts to understanding the validity of this approach in the literature. Although there have been a few studies [2, 8, 39], evolutionary algorithms are generally adopted off-the-shelf to simulate socio-economic learning. Without a methodical adoption of algorithms for simulation, any socio-economic interpretation of the process and results thereof, lack the rigour necessary to confidently say that the interpretations are indeed sound. In fact, if one does not understand the algorithms, one cannot understand what they simulate. Given that co-evolution is a co-adaptive search process, it is not necessarily clear what connects search and simulation. If agent based simulations are to be seen as search processes or processes of adaptation, then one needs to reconcile search and desired goals from simulations.

The fact of there being mis-matches in socio-economic simulations using co-evolution are remarkably similar to the fact that mis-matches are often seen to occur in co-evolutionary algorithms, where they have been shown to occur due to the lack of rigour in the definition of the co-evolutionary solution concept [53]. Our aim is to discipline ACE research such that, ideas from co-evolutionary algorithm design research, more precisely, the notion of solution concepts in co-evolution, are *not overlooked* in this field growing largely independently of co-evolution.

Game theory gives us useful idealised results in terms of what outcomes one may expect with players interacting under theoretical assumptions. Although these theoretical assumption are far too strict and do not in any way model the reality of the particular interaction or game under question, they are indeed required for a closed form solution to the game. Although the value of these theoretical outcomes is arguable, at least in economics and psychology research, for the purpose of understanding co-evolution and as a consequence, agent based simulations, they are indeed a boon. These solutions give us a goal to aim for, without which, evaluating the outcomes from a co-evolutionary process becomes extremely hard, or even impossible, for the very fact that co-evolution is an open ended or even a cyclical (e.g. red queen effect [33]) process. To put a handle on the process, one needs a definition of the quality of the solutions one expects from the process. Game theoretic solutions give us these quality attributes which we may use in order to engineer (analyse and refine) co-evolution towards achieving

them in the solutions that emerge out of the process.

We essentially want to make sure if we can 'engineer' co-evolution in a methodical manner, specifically utilising the framework described in Section 3.2, towards modelling socio-economic situations such that, pre-defined or desired outcomes from the socio-economic situation considered, are indeed achieved or achievable, or at least that the mis-interpretations of the simulations in socio-economic terms (or mis-interpretations of socio-economic terms within simulations) be avoided. Following this, we can then also delve deeper into understanding and analysing how the co-evolutionary process fairs in achieving outcomes (for example, deviations from rational equilibrium outcomes, which may relate to real world outcomes with humans) that may have been explained to have come about as a result of the mis-interpretations, enabling the formation of closer matches between simulation and reality, if need be.

3.5 The Crux of the Problem: Forcing a Socio-economic Interpretation

3.5.1 Preliminaries

In this section we describe the game considered in the thesis, and the co-evolutionary algorithm and settings previously used in [59].

3.5.1.1 Alternating Offers Multiple Issue Bargaining Game

We consider the multiple issue, finite horizon (finite number of rounds) version of Rubinstein's [127] alternating offers bargaining game. There are two agents, 'Player 1' and 'Player 2', and Player 1 always starts the game. Each agent (as an *offerer*) in turn proposes an offer (a division of the surplus on m issues, expressed as a vector $\vec{o}$) to the other agent (i.e. the *responder*). The responder decides whether or not to accept this offer. This decision is made by matching the *utility* of this offer against a threshold. Each round of the game consists of an offer proposed by the offerer and the decision on the acceptance or rejection of it by the responder. There are a maximum of n rounds. At the end of each round,

the game terminates with a probability $1 - p$. The probability p (probability that the game proceeds to the next round) reflects the uncertain nature of real world bargaining interactions, from the point of view of the agents not necessarily knowing the actual deadline of the game. The game ends in a disagreement if there is a breakdown (which, as said, occurs with probability $1 - p$) or if the responder rejects the offer in the last round. In either case both agents receiving nothing (i.e. the utility is zero). The game ends in an agreement if the responder accepts the offer in any of the rounds before the game terminates. In this case, the agents receive a positive payoff decided by their respective utility functions. Refer to Fig. 3.1 for a diagrammatic description of the game. As in [59], we

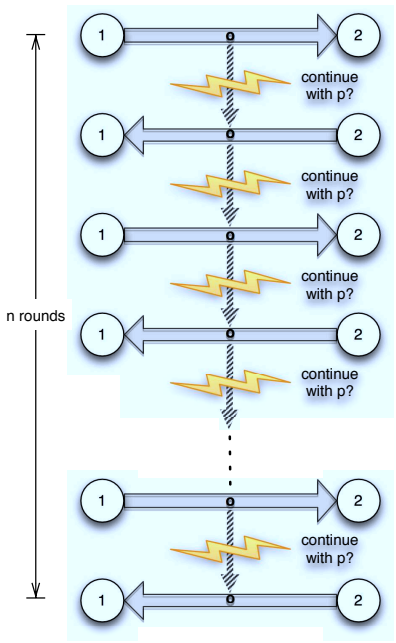


Figure 3.1: Alternating offers multiple issue bargaining game.

assume, without loss of generality, that the total bargaining surplus available for each issue is *unity*. For either agent, the utility of an offer $\vec{o}$ is the dot product $\vec{w_j}.\vec{o} = \sum_{i=1}^{m} w_j^i o^i$, where $\vec{w_j}$ is a vector containing agent j's weights for each issue.

A particular instance of the game is defined by specifying values for the number of issues, m, the probability of going on to the next round, p, the maximum number of rounds, n, and the the weight vector $\vec{w_j}$ for agent j. We consider

fixing the number of issues (m) to 2 to begin with, since the majority of the results reported in [59] consider this value of m. We also consider fixing $\vec{w_1}$ to $(0.7, 0.3)$ and $(\vec{w_2})$ to $(0.3, 0.7)$, again because the results reported in [59] consider these. Note that the weight vectors $(\vec{w_j})$ are independent of the offers made by an agent or the offers received from the other agent. An agent only gets to use its respective weights when calculating the utility of an offer, for both the offer made by itself and the offer received from the other agent. Other instances of the game are specified as and when necessary in order to explore the answers to the main questions raised in the thesis. Thus, a specific value for p and n instantiates the game (see Table 3.1). For each combination of p and n, there is a unique SPE solution that can be calculated using backward induction (as discussed in Chapter 2). It is this equilibrium point that characterises the optimal strategies the co-evolutionary algorithm should evolve.

As an example, we consider the game with no breakdown i.e. $p = 1$. In this case, the game theoretic solution (SPE) is the obvious one, that if the game is of 'odd' length (n is odd), the first agent gets everything on all issues (payoff of 1) and the second gets nothing (payoff of 0). On the other hand, if the game is of 'even' length (n is even), the first agent gets nothing and the second gets everything on all issues. In other terms, if the number of rounds is fixed, then the agent that turns out to be the responder in the last round should accept any offer since the alternative is disagreeing, hence receiving no utility at all.

3.5.1.2 The Co-evolutionary Algorithm

We apply the same co-evolutionary self adaptive $(\mu + \lambda)$-Evolution Strategy (ES) used in [59]. Two populations (of size μ) are considered, one for each type of agent (Player 1 or Player 2). Each individual in a population plays a game with every individual in the other population, and the fitness is evaluated as the average utility that the individual gets from all the games. At each generation, λ individuals are selected from each population uniformly at random (independently) and then mutated to form the two offspring populations (of size λ), one for each agent type. The individuals in the offspring populations are evaluated by playing against every individual in the opponent parental population. The fitness of each

Table 3.1: Parameter settings for the game and co-evolutionary $(\mu + \lambda)$-ES.

Game Parameters	Weights of agent 1 ($\vec{w_1}$)	$(0.7, 0.3)$
	Weights of agent 2 ($\vec{w_2}$)	$(0.3, 0.7)$
	Number of issues (m)	2
	Continuation probability (p)	specified as needed
	Maximum number of rounds (n)	specified as needed
EA Parameters	Parental population size (μ)	25
	Offspring population size (λ)	25
	Selection scheme	$(\mu + \lambda)$-ES
	Mutation model	self-adaptive
	Initial standard deviations (σ_i)	0.1
	Minimum standard deviation	0.025

individual is the average utility obtained by playing against these opponents. The μ fittest individuals are selected from the respective pools of $\mu + \lambda$ individuals of both types for the next generation[1]. Fig. 3.2 shows a schematic diagram of this process. Table 3.1 shows the parameter settings for the algorithm used in [59] and that we will replicate.

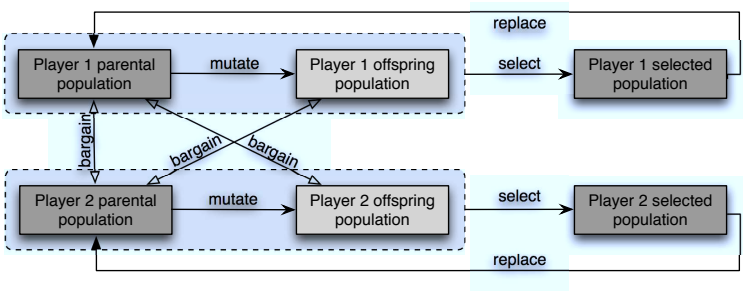


Figure 3.2: Co-evolutionary $(\mu + \lambda)$-ES.

The strategy representation used is the same as in [59], and shown in Fig. 3.3. The agent strategy specifies the offers $\vec{o_j}(r)$ and thresholds $t_j(r)$ for each round r in the game for agents $j \in \{1, 2\}$. As such, the strategy is specified by a string of real numbers in the interval $[0.0, 1.0]$. At the start of each run, these offers and thresholds are initialised by sampling random numbers in the interval

[1]This scheme resembles the *individual learning* notion of simulating socio-economic learning within ACE.

$[0.0, 1.0]$ from a uniform distribution. Moreover, the mutation of an individual happens as follows. Each real value, say x_i (which is a value either representing the offer on one of the issues or a threshold), in this string that specifies the strategy, is mutated by adding a zero mean Gaussian variable having a standard deviation of σ_i to x_i, giving x_i', i.e. $x_i' = x_i + \sigma_i N_i(0,1)$[1]. The standard deviations (σ_is) decide on the step size of the mutation, and are associated with each x_i in the representation. These σ_is are self-adaptive in the sense that, the mutation operator in this algorithm first updates σ_i according to $\sigma_i = \sigma_i exp[\tau' N(0,1) + \tau N_i(0,1)]$[2], and then uses this updated σ_i to mutate the corresponding x_i, giving x_i'.

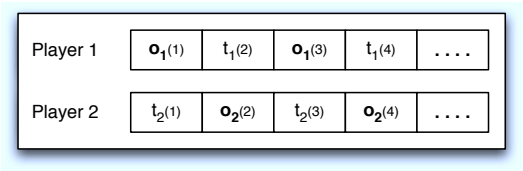


Figure 3.3: Strategy representation.

3.5.2 Bounded Rationality Leading to Divergence

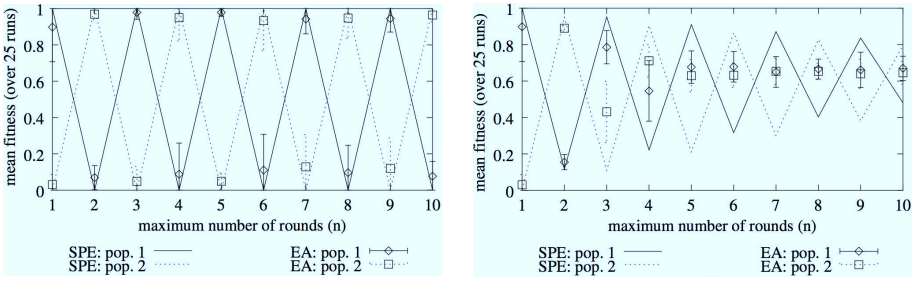
We consider the problem of co-evolving optimal agents for the *alternating offers bargaining game* [127]. Previous work [149] using co-evolution for finding the theoretical equilibrium (SPE) for the game suggested that it was not possible to achieve optimal agents at the end of the co-evolutionary process, hence, that the algorithm did not converge.

Fig. 3.4 shows the mismatches between evolutionary and SPE results for bargaining games with $p = 1$ and $p = 0.95$, as reported in [59]. Lines join SPE solutions to guide the eye. It can be seen that the mean fitness of the populations over 25 runs is somewhat close to the SPE values but does not converge to them.

[1] $N_i(0,1)$ is generated afresh for each x_i.

[2] τ' and τ are specified as $(\sqrt{2l})^{-1}$ and $(\sqrt{2\sqrt{l}})^{-1}$ respectively [17]. The zero mean Gaussian variable $N_i(0,1)$ associated with τ is generated afresh for each σ_i, whereas $N(0,1)$, which is associated with τ' is only generated once and applied to $\vec{\sigma}$ (i.e. applied to all σ_is associated with x_is of an agent).

This holds for any p or any n. The reasons are attributed to the agents being *"myopic"* [59], in that the agents had no memory to remember past interactions, no explicit rationality principles to use, and for more realistic game settings with a stochastic element defining when the game ends, they were supposedly unable to reason backwards from the deadline. While these issues may be interesting, they do not directly answer why co-evolution, in this case, does not lead to optimality and convergence. Since a given setting of the bargaining game has only one SPE, it is worth investigating whether a co-evolutionary algorithm can evolve the agents' strategies towards optimality. In Section 3.6, we further analyse the co-evolutionary algorithm used in [59, 149] to understand why optimal agents cannot be co-evolved and whether there are better co-evolutionary approaches that can be used in this case. The goal here is to identify general issues that may influence co-evolution, not just for a particular game considered. Since this myopia is reported for all values of p and n, it seems safe to look into the most simple setting of the game, that of $p = 1$, in our following analysis.

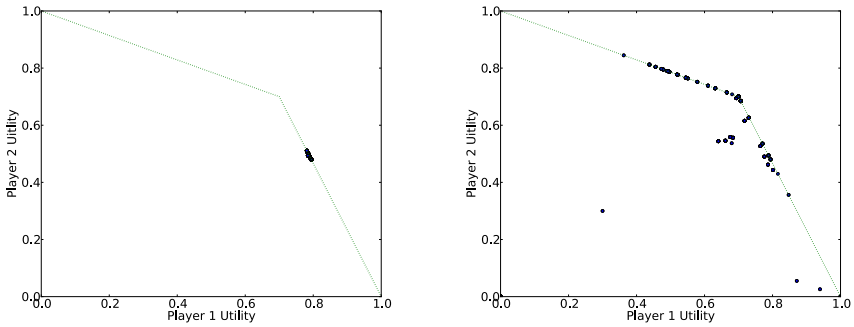


(a) $p = 1$ [59]. (b) $p = 0.95$ [59].

Figure 3.4: Comparison of evolutionary results with SPE results for $p = 1$ and $p = 0.95$ [59].

We re-implement the methodology from [59]. When running the algorithm we see that, independently from the values of p and n, the populations indeed evolve towards the pareto-efficient frontier (Fig. 3.5). And, as reported in [59], the agents keep exploring the search space as a *"moving cloud"* of agreements along the frontier, instead of evolving towards the equilibrium. Moreover, we see that, the populations scatter in the long term, only to regroup, move along the frontier and scatter again. Fig. 3.5 depicts the described "breakdowns". This

should not happen if our solution concept is that of finding the equilibrium for the game, especially with elitist selection.



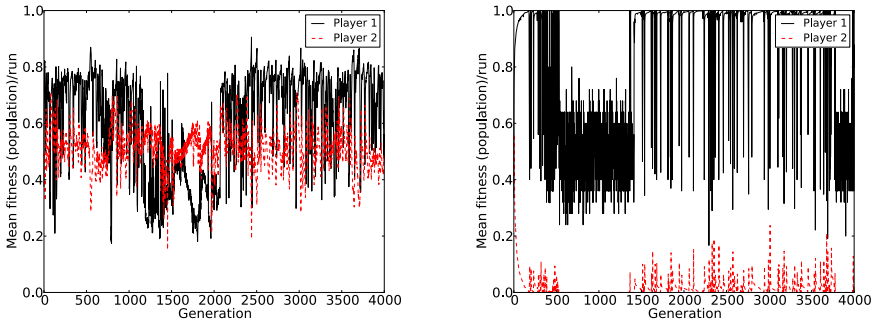
(a) After 3200 generations. (b) After 3700 generations.

Figure 3.5: Outcomes of games played (agreements reached) by both parental populations for $p = 0.7$ and $n = 10$ from a typical run. $SPE = (0.769, 0.538)$. The dotted line is the pareto-efficient frontier.

In Fig. 3.6 we plot the mean fitness of the populations in a typical run for $p = 0.7$, $n = 10$ (Fig. 3.6(a)) and $n = 1$ (Fig. 3.6(b)) respectively, $p = 1$ by default in the latter. In the first case we can see that the mean fitness of both populations just goes up and down. The figure suggests that the strategies are not evolving towards any direction. The wrong co-evolutionary solution concept might have thus been applied. From Fig. 3.6(b) we can understand a bit more. Since for the simple game with $n = 1$, the optimal solution is 1.0 for Player 1 and 0.0 for Player 2 (as discussed in Section 3.5.1.1), the fitness trajectories in Fig. 3.6(b) are clearer. We see that the mean fitnesses do head towards the optimal values and then collapse repeatedly. Although the causes for this divergence still remain unclear, we see that the co-evolutionary methodology does not converge even for this simpler game. In Section 3.6, we continue studying the game where $p = 1$ and simplify the experimental setup in order to gain insight into the co-evolutionary dynamics, specifically convergence.

In particular, we remove the influence of a population by considering only 1 individual per agent type in the co-evolutionary setup (i.e. we get the (1+1)-ES[1]).

[1]Note that in a (1+1)-ES, the offspring replaces the parent only if its fitness is greater than or equal to that of the parent (i.e. if the parent and offspring have the same fitness, selection



(a) $p = 0.7$, $n = 10$. (b) $n = 1$.

Figure 3.6: Mean fitness of the populations in a typical run.

The idea is to study the causes of the breakdowns by looking at the individual strategies. All other parameters remain the same as in the original population based algorithm.

3.6 Possible Solution: Engineering Co-evolution

3.6.1 Co-evolutionary Solution Concepts

A solution concept is a notion borrowed from game theory and specifies whether a strategy played by an agent adheres to a set of standards that make the strategy preferable to be used over other strategies, and thus, retained. It separates solution strategies from those that are not, almost like a filter through which the whole search space of strategies is passed, retaining only the desired strategy. However, designing this filter is no mean task. We do not have the luxury of explicitly scrutinising each strategy in the search space, which leads to sampling strategies, evaluating them, and following a gradient. The solution concept can thus be specified as using the gradient to separate potentially preferable strategies from those that are not. Co-evolution however, makes finding this gradient harder, because the evaluation of strategies is relative to the strategies sampled, and not a static/absolute function to optimise on.

The pressure towards filtering out the desired strategy, is built on by channel-

is not performed uniformly at random).

ing the solution strategies along a certain path in the incomplete and changing view of the search space. The idea is for the incomplete and changing view of the search space to be timely for it to be useful to guide the solutions towards the desired solutions. This is very important in co-evolution because the incomplete view of the search space is reactive to the solutions representing the current state of the algorithm. A secondary search problem thus results, that of finding who to interact with, given that the solutions themselves present the incomplete view of the search space. The opponents that describe the incomplete yet useful view of the search space to adapt with, need attention in their discovery and use.

Thus, co-evolution involves solving a secondary search problems: one that allows for selection of preferable strategies, and the other that allows for the selection of a sample of strategies against which evaluation of any strategy may provide a preferable gradient. *Designing and following the primary and secondary search problem gradients entails a thorough investigation of the elements that make up a co-evolutionary algorithm, and is what we call the* ***implemented co-evolutionary solution concept*** *in this thesis.*

This includes the representation that decides the nature of the search space, interaction, evaluation and selection schemes that decide on the nature of the sample, the gradient provided by the sample, and the manner in which the gradient is followed, and the variation operators that help traverse the search space. In order to use co-evolution to lead to a certain kind of behaviour one has to decide on these elements for the co-evolving entities, giving us the implemented solution concept. The discussion in Section 3.5.1.2 describes the implemented co-evolutionary solution concept used in previous work [59]. So, given an expected view on what co-evolution should result in, any mismatch in the actual behaviour of the algorithm (for example, expecting convergence to the equilibrium outcomes but the algorithm resulting in outcomes diverging from the equilibrium in previous work [59]) signifies a mismatch between the desired solution concept (desired strategise to be filtered out) and implemented solution concept.

Note that, if the idea is to simulate 'real' economic behaviour, one has to tune the aforementioned elements first in a methodical manner, such that they do indeed do so, before interpreting the results obtained from the algorithm. Otherwise, the conclusions drawn from the incomplete treatment of these elements are

at best premature. This is a problem still overlooked in ACE literature and one of the aims of this thesis is to emphasise the importance of co-evolutionary solution concepts within the field via the application of our methodical framework, the first integral part of which is described in Section 3.2.

3.6.1.1 Relationship Between Divergence in Co-evolution and Divergence Leading to Misinterpretations in ACE

Co-evolution has its share of problems in there being a mismatch between the desired and implemented solution concepts [53]. Using co-evolution directly off-the-shelf in ACE does not solve these problems of course. It makes sense to assume that these problems will remain to be understood in depth, instead of assuming for them to be an 'economic feature' (as opposed to an 'algorithmic incompetence') and interpreting or explaining the problems as socio-economic terms. ACE often sees such problems as a *"source for additional insights about the fine-structure of the emergence"* [40] within socio-economic simulations caused by socio-economic phenomena interpreting various aspects of these simulations. For example, the problem of divergence from the equilibrium solutions, may be seen as due to certain aspects of the algorithm which could be given a socio-economic meaning, and this socio-economic meaning can in turn be then seen as the source of additional insights about the nature of emergence within simulations. When making socio-economic assumptions about the algorithm however, if the insights are not sourced (i.e. scrutinised), but instead the fine structure of emergence simply taken for granted (in other words, explaining divergence being caused due to some assumed socio-economic phenomenon like bounded rationality, instead of finding the algorithmic source of the divergence), we cannot be sure whether or not it is indeed algorithmic incompetence that mars the simulations. As such, socio-economic phenomena may get mis-interpreted as leading to these fine structures of emergence, the case in point being bounded rationality getting mis-interpreted as causing divergence within simulations. We are thus of the opinion, that working on the co-evolutionary solution concept should not be overlooked, or else the use of socio-economic phenomena of interest for explanations within the simulations may not be fully justified.

3.6.1.2 What Should Explain Divergence?

As explained previously in Section 3.5, the divergence between expected and observed outcomes from the co-evolutionary simulations were attributed to 'bounded rationality'. Not working on the co-evolutionary solution concept to see if this divergence can be overcome, and explaining the lack of rigour in understanding the process by holding bounded rationality of the agents responsible for it alone, does not do justice to either field, i.e. co-evolutionary algorithm design on the one hand, and ACE on the other. If we could come up with an algorithm (a refinement of the co-evolutionary solution concept implemented previously, described in Section 3.5) which result in agents converging to the equilibrium solutions for the games considered, the real meaning of bounded rationality may come to surface. We will essentially come to know what bounded rationality does not mean in computational or algorithmic terms, and indeed come to know what it was being assumed to represent (algorithmically). If, on the other hand, we are unable to engineer an algorithm that does this, we would still delve deeper into understanding the co-evolutionary process and its relationship with observed outcomes in computational or algorithmic terms (as opposed to unassessed socio-economic terms), which we may then consider seeing as bounded rationality (i.e. if there seems to remain no reason for such a consideration to get challenged), though in explicit algorithmic terms (as opposed to current practice in ACE of simply assuming bounded rationality to be present in the algorithms from the outset). One could then even envisage quantifying the socio-economic phenomenon of bounded rationality as an explicit computational or algorithmic entity to understand how it really affects co-evolution towards specific outcomes, something we do consider in Chapter 4.

3.6.2 Convergence to Equilibrium

We introduced this problem of forcing an economic interpretation on co-evolutionary simulations by taking the case described in [59], where bounded rationality is seen as the cause for all the observed deviations from optimality (deviations from the equilibrium). We want to understand if it is really bounded rationality or something else, that causes the deviations. In other words, we want to understand

why the co-evolutionary algorithm does not converge to optimality, which in our case is the equilibrium. We see bounded rationality as a premature explanation of the deviations, since a thorough investigation on what it really means in terms of how it relates to the co-evolutionary search process was not carried out. The question that immediately follows is whether or not the implemented solution concept can be engineered so that we achieve convergence to the equilibrium with boundedly rational agents? The following analyses and refinements of the algorithm, applying our methodical framework from Section 3.2, sheds light on this issue.

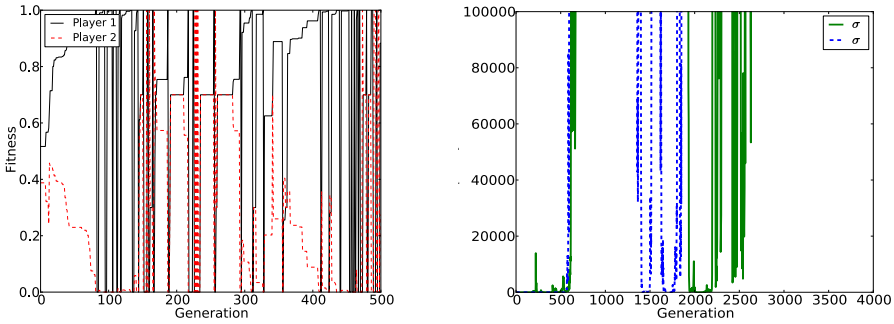
3.6.3 Working on the Co-evolutionary Solution Concept for Simple Games (p=1)

3.6.3.1 Variation

Fig. 3.7(a) shows 500 generations of a typical run. We see that (as also noted for the single issue case in [149], albeit, with a population), the individuals actually reach the SPE of $(1.0, 0.0)$ for odd n and $(0.0, 1.0)$ for even n. This however should not happen. It is theoretically impossible to reach the optimum $\vec{x^*}$ of a continuos optimisation problem in finite time, but only a point $\vec{x}$ such that $|\vec{x} - \vec{x^*}| \leq \epsilon$ for any constant $\epsilon > 0$ [17]. The ES should converge towards the optimum by gradually decreasing the step size $\vec{\sigma}$ and only reach the exact optimal point at infinity. Hence, there seems to be a problem with the algorithm implementation.

It turns out that the mutation operator sets any mutated value larger than unity (or smaller than zero) to unity (respectively zero) [59], instead of considering solutions outside the search space as infeasible. Although this pushes the agents' strategies to optimality (i.e. for $p = 1$), it turns out to be destructive. Once the agents reach the SPE values exactly, the responder becomes indifferent between *getting nothing* and *disagreeing* with the opponent. There is no selection pressure for the responder to keep a threshold of 0.0 (while there would have been for $0 + \epsilon$ for any positive ϵ). Hence the responder ends up accepting a random threshold in the following generation with a high probability (in fact, 1) resulting in a disagreement (i.e. the fitness goes down to 0). The new threshold would not be *too* bad if the mutation step size were decreasing in time as it should. However,

the step size $\vec{\sigma}$ ends up evolving in the opposite way. The higher the $\vec{\sigma}$, the higher is the probability that an infeasible solution is created which in turn is transformed into the optimal one. In Fig. 3.7(b) we plot the $\vec{\sigma}$ values for Player 1 in a typical run. The same phenomenon happens for Player 2. Where the values of σ_i's are not visible, they are higher than the plotted extreme.



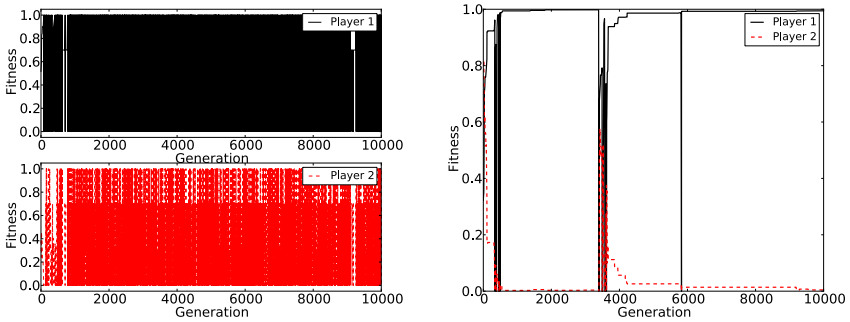
(a) A typical run. (b) σ_i for Player 1.

Figure 3.7: A typical run of the (1+1)-ES using the mutation procedure in [59] for $n = 1$.

Forcing infeasible players to be equilibrium players or clipped to the bounds of the search space does not do justice to the co-evolutionary search, i.e. mis-guides the search. The algorithm cannot follow its perceived gradient towards optimal solutions, as it is made to jump out of bounds and achieve infeasibility disguised as optimality. The self adaptive step sizes are testimony to this behaviour of the algorithm as they get extremely large, forcing the solutions towards infeasibility. Moreover, being forced to optimality exactly, further misguides the algorithm once optimality is achieved. If we are to engineer a solution concept with having the ability to construct the necessary gradients towards optimality, once we reach optimality, the algorithm should stay put instead of diverging. The fact that a disagreement and getting nothing result in the same fitness for the responder i.e. there is no gradient that the algorithm constructs over the threshold of the responder once optimality has been achieved, divergence is bound to happen. As said before, the ES should converge towards the optimum by gradually reducing the step size and only reach the exact optimal at infinity, which would happen if instead of accepting infeasible solutions, the gradients are properly constructed

and used by the algorithm so that infeasible solutions are indeed considered infeasible. We emphasise that this problem with the algorithm clearly suggests that bounded rationality should not be considered as an explanation of divergence from optimality yet.

We modify the algorithm such that infeasible solutions (i.e. solutions outside the search space) have worse fitness than any feasible search space point (i.e. -1.0), hence are never accepted.



(a) A typical run. (b) Worst (most divergent) of 25 runs.

Figure 3.8: Runs for the (1+1)-ES using (a) the mutation procedure in [59] (separate plots for agent types to make the severity of fluctuations unambiguous) and (b) the corrected version, for $n = 1$.

In Fig. 3.8, the worst run of the corrected (1+1)-ES is compared with a typical run of the original version. It clearly shows the difference this simple correction to the original approach makes. Now the strategies evolve towards optimality for long periods of time. However, breakdowns still occur (although far less frequently) and become more severe as the number of rounds n increases.

3.6.3.2 Interaction

We call a pair of strategies as *incompatible* if they disagree when playing with each other. A closer look at the evolved strategies reveals that the simultaneous selection of *incompatible* offspring as the new parents leads to a disagreement, and hence, a collapse. To illustrate this, we call P_1 and P_2, the two parents (one for each agent type) at any given time during the co-evolutionary process, and the offspring, C_1 and C_2 respectively. If (P_1, P_2) are compatible, (P_1, C_2)

are compatible, and (C_1, P_2) are compatible, it does not necessarily follow that (C_1, C_2) will be compatible. Figure 3.10 illustrates this. We argue that this was going un-noticed in the original work [59] and being attributed to the *myopic* or boundedly rational properties of the agents. Note that an agent in equilibrium can be myopic in the sense that it may only know how to play against another agent in equilibrium, which is when the issue of myopia should be scrutinised. This is not the case here. Having not played a game before, if the offspring are incompatible and selected to be in the next generation, then they will inevitably disagree. This problem (i.e. *selection of incompatible opponents*) does not depend on *myopic* properties of agent strategies, but on an unusual implementation of the co-evolutionary solution concept.

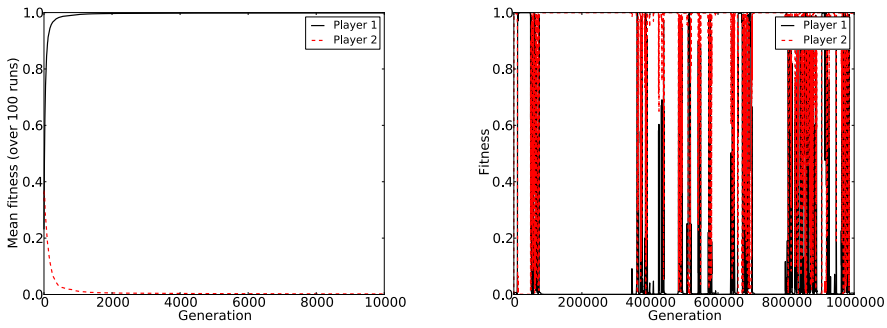
If we imagine the parents to be close to the equilibrium, it can so happen that the offspring of both adapt towards fitting the parental opponents in such a way that the offspring of Player 2 increases its threshold but receives the same fitness when evaluated against the parental opponent. At the same time, the offspring of Player 1 fits on to the threshold of Player 2 (parent), leading to an increase in fitness. Both the offspring will be selected for in this case and they will disagree. If offspring interact before selection, the gradient towards optimality may not be compromised. If the offspring of Player 2 interacted with the offspring of Player 1 for further evaluation for its worth as providing a useful secondary search problem gradient (can also be looked at from the other player's point of view i.e. if offspring of Player 1 interacted with offspring of Player 2 for further evaluation for its worth as providing a useful secondary search problem gradient) the breakdown could be avoided. Playing with opponents which show a gradient towards convergence can help tackle the primary search problem of finding the optimal. Knowing that these opponents are indeed useful, their usefulness coming from the gradient provided by the players in turn, can help tackle the secondary search problem. The co-evolutionary solution concept can, in this manner, be refined. Note that the nature of the problem is such that the primary and secondary search problems, on seeing from the point of view of the other side or player, change roles. The primary search problem for Player 1 is the secondary search problem for Player 2 and vice versa.

In the following, we modify the co-evolutionary solution concept, i.e. the inter-

action element, by allowing the offspring to play games against each other before selection. If this game results in a disagreement, the offspring are not accepted (they get a negative fitness of -1.0), otherwise, the co-evolutionary process carries on as in the previous co-evolutionary solution concept. This, by definition, eliminates the problem of *selection of incompatible opponents* altogether. Fig. 3.9(a) shows the mean fitness across 100 runs of the modified (1+1)-ES. No divergence was seen in any of the runs (we also allowed the algorithm to run for a million generations, without seeing one breakdown). Since there is no reason for breakdown occurring if we add populations, the algorithm will now work also for $\mu > 1$ and $\lambda > 1$.

As we can see, the co-evolutionary solution concept has clearly not been thoroughly scrutinised in previous work, but instead a premature explanation of the divergence as caused by bounded rationality is given instead. It is clear that bounded rationality is not necessary to explain the divergence at this point, as there is a simpler explanation in the form of a modified solution concept, which promises to lead to convergence to optimality.

However, the modifications we have applied do not imply convergence for $n > 1$. In fact, already for $n = 2$, we observe some fitness breakdowns again. Figure 3.9(b) shows a run of one million generations for $n = 2$.



(a) $n = 1$. (b) $n = 2$.

Figure 3.9: Modified (1+1)-ES for (a) $n = 1$ (mean across 100 runs) and (b) $n = 2$ (a typical diverging run).

Zooming into Fig. 3.9(b), to the generations where there is a sudden change from the convergent trajectory (generations starting from 374400 for instance), and investigating the agent strategies in these generations, reveals the cause for

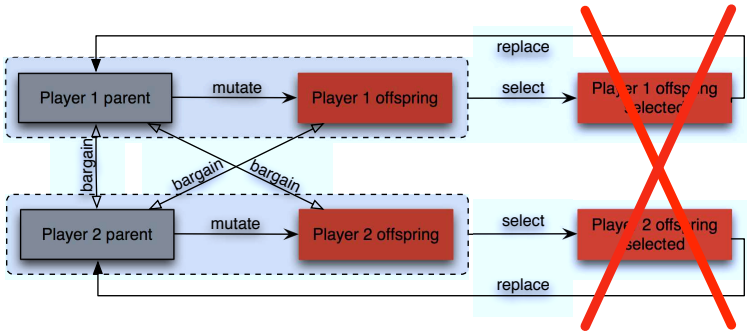


Figure 3.10: Selecting incompatible opponents.

these breakdowns. Fig. 3.11 takes the two agent (P_1 and P_2) strategies apart and shows what each agent receives and wants in terms of utilities in both rounds (since $n = 2$). It can be seen that the *evaluation of a strategy* is *local* with respect to the rounds. Since the fitness of an individual is computed whenever there is an agreement, which can happen in any round, the strategies only consider the agreement round to evaluate themselves (i.e. it is the offers and thresholds of the agreement round that determine the strategies' payoffs). In Fig. 3.11, the agreements happen in round 2 to begin with (when the outcomes are near SPE). There is no real selection pressure for modifying round 1 offers and thresholds, so the strategy chunks corresponding to this round vary randomly and at some point are such that P_1 offers more than what P_2 needs. Consequently, the agreement happens in round 1. Now, P_1 can take advantage of the fact that it is in the driver's seat (as the utility of both agents is decided by the offer it makes, when this offer is accepted, which it is), while P_2 has lost its selection pressure towards higher fitness (mismatch between its threshold and P_1's offer leading to P_2 getting what it is given). Meanwhile, the selection pressures for offers and thresholds in round 2 vary randomly since the players agree in round 1 and never reach round 2. By modifying the fitness evaluation procedure such that it credits the whole strategy instead of just the part corresponding to the agreement round, we may solve the problem. Alternatively, had the representation not been treating each round as being separate from each other, we may solve the problem too.

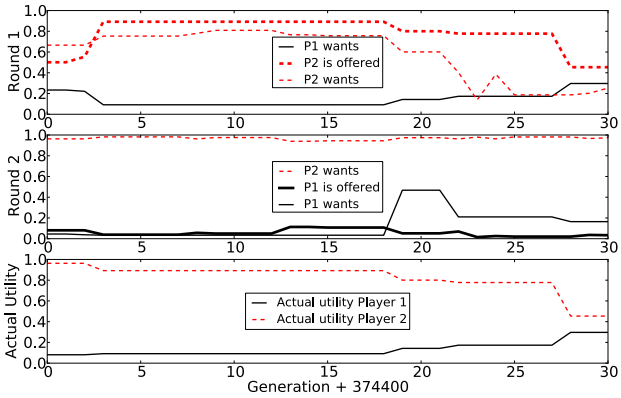


Figure 3.11: Snapshot of the modified (1+1)-ES for $n = 2$. Strategy evaluation is "local" with respect to the rounds.

3.6.3.3 Evaluation

In order to make evaluation of strategies less local, we introduce a 'sortedness' property that any newly generated strategy should adhere to. This property suggests that for a game where $n > 1$, given the original strategy representation, we consider a strategy feasible (of positive fitness, that results from the outcome of the game) if and only if, the values that define the entire strategy monotonically decrease with the number of rounds. In other words, the concessions that a strategy represents monotonically increase with time. The reason why this property should cater for the evaluation being less local than the previous methodology is that it is now clear that if there is an agreement at any round in the game, the agreement is lead to this round through previous rounds where the concession was not as much as in the current round. This is in contrast to the previous setup where concessions could be random across rounds, thus, not adhering to a global structure where a round does not inform (*implicitly* by way of evaluation) the previous or the next round. The strategy is thus not allowed to vary randomly in rounds where the agreement did not happen, as was the case previously, which lead to the switching of the agreement round, which further lead to divergence from equilibrium. Thus, there is now some handle on the local loss of selection pressure by way of relating the rounds through this monotonicity principle.

The way the evaluation works now is as follows: At each evaluation step (when

two individuals play a game with each other), we introduce two rules that they need to adhere to, failing which, they are made infeasible. These rules are:

Rule 1: A player's threshold in the next round should be lower than or equal to its current offer's utility.

Rule 2: A player's offer's utility should be lower than or equal to its threshold in the previous round.

Together, the above two rules guarantee that an individual is sorted until the round it agrees on with an opponent or until the last round if there is a disagreement (including there being an agreement in the last round). If however, the opponent fails to adhere to these rules, then the individual cannot be evaluated and receives zero payoff. Moreover, we do not want both offspring to be selected for at the same time to be in the next generation and find out that they are not sorted until their agreement round (in case they agree - which is the only way they can indeed be in the next generation, given our modification for *selection of incompatible opponents* in Section 3.6.3.2). The offspring which violate these rules are hence penalised.

The reward structure of the algorithm now is such that, if there is an agreement and the player being evaluated does not violate the rules, it gets a positive payoff (payoff from the game), provided the opponent it is being evaluated against does not violate the rules. If the opponent violates rules but player does not, the player gets zero payoff (since being evaluated positively against an infeasible solution would not be right). The player receives a negative payoff of -0.5 if it violates the rules. If the player is an offspring and it is incompatible with the opponent offspring, both receive a negative payoff of -1.0. A player which is out of bounds (refer to Section 3.6.3.1) receives a negative payoff of -2.0. This payoff structure is there to demarcate the problems so they do not interfere with each other. For instance, we do not want an out of bounds player to replace a player that violates the rules but which is still within bounds. Having the same negative payoff can make this happen.

Note that the problems discussed so far for equilibrium selection in games with multiple issues, translate directly to equilibrium selection for games with a

single issue. Infeasible solutions are in fact more likely to appear in the single issue case as the strategy is composed of a smaller set of real numbers (hence the problem identified in Section 3.6.3.1 is indeed present). Incompatible opponents are more likely to appear than in the multiple issues case, as the incompatibility arises when individuals disagree, which, for a game with n rounds, happens when there is no agreement, even at round n, and multiple issues games introduce a wider range of possibilities of agreements if the game indeed contains win-win agreements as possibilities (hence the problem identified in Section 3.6.3.2 is indeed present). So, in order to simplify our analyses, we consider bargaining games with a single issue from now on (and in the thesis from this point on). Note that the essential problem with the co-evolutionary solution concept alluded to in Section 3.5, which considers the multiple issue bargaining game, is unchanged for a game with a single issue. The latter has also been dealt with in a similar fashion to equilibrium selection in multi-issue bargaining perviously [148, 149], hence our analyses applies to equilibrium selection in single issue bargaining games equally. Be it a single issue or a multiple issue case, the equilibrium solutions for either, for the case of $p = 1$ and $n \geq 1$ are the same, i.e. last offerer gets everything whereas the last responder gets nothing.

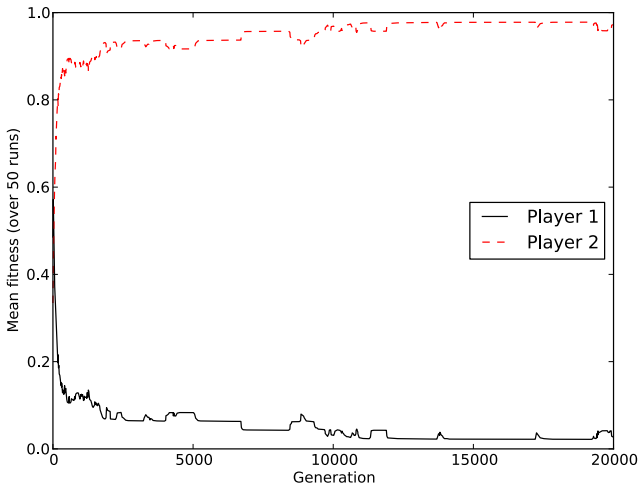
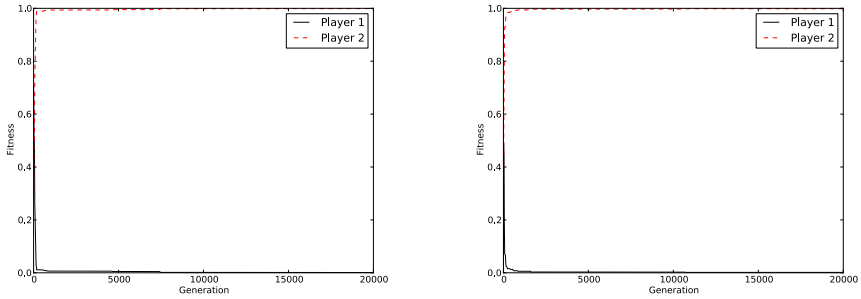
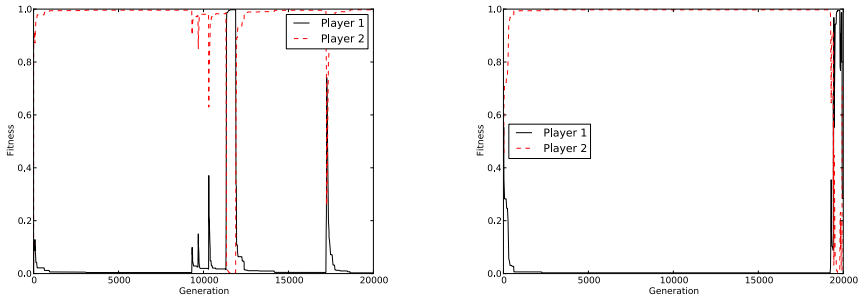


Figure 3.12: Mean fitness across 50 runs of the algorithm with the 'sortedness' rules in place, for $n = 2$.



(a) Converging run 1. (b) Converging run 2.

Figure 3.13: Converging runs for $n = 2$ with the 'sortedness' rules in place.



(a) Diverging run 1. (b) Diverging run 2.

Figure 3.14: Diverging runs for $n = 2$ with the 'sortedness' rules in place.

Figure 3.12 shows the mean fitness across 50 runs of the algorithm with the 'sortedness' property in place, for $n = 2$. One can see the tiny blips along the course of evolution as the algorithm attempts to converge to equilibrium. Though there are runs which do indeed converge to equilibrium, as can be seen in Figure 3.13, divergence is still a problem. If we single out the runs that cause these blips, we can say more. Figure 3.14 shows the singled out runs and it is clear from them that the algorithm does not converge. The reason for divergence is that there is still a possibility of a switch from an agreement happening in one round to another. If ever the agreement happens in an 'odd' round, the outcomes favour Player 1, otherwise they favour Player 2. This happens due to the fact that the offerer in any round decides the payoff for each player from the game, provided the agreement happens in that round. On the other hand, the responder finds itself on a fitness plateau as no matter what its threshold, the payoff is always going to be the same and as governed by an agreement happening for some value of the

offerer's offer, which must necessarily be greater than the responder's threshold. The following example illustrates when a switch is possible, and hence when divergence occurs.

Table 3.2: Player strategies per round for $n = 2$.

	Round 1	Round 2
Player 1	$\mathbf{o}_1(1)$	$t_1(2)$
Player 2	$t_2(1)$	$\mathbf{o}_2(2)$

As a simple example, let us consider a linear relationship between utility functions of the two players such that $u_2 = 1 - u_1$. An easy way to think about this relationship is if we consider a single issue with our type of utility function (weighted sum), where the preference weights for the single issue for both players are 1.0 i.e. $u_i(o_i) = o_i$, so $u_1 = 1 - o_2$, as $u_1 = 1 - u_2$. Table 3.3 details the possible utilities for players in each round for a single issue bargaining game, with $n = 2$.

Table 3.3: Player utilities per round for $n = 2$, for a single issue bargaining game.

	Round 1	Round 2
Player 1	0 or $o_1(1)$	0 or $1 - o_2(2)$
Player 2	0 or $1 - o_1(1)$	0 or $o_2(2)$

Since $t_2(1) \geq o_2(2)$ from the sortedness property, we know that $1 - t_2(1) \leq 1 - o_2(2)$. This suggests that there are 3 possible relationships between $o_1(1)$ and $t_2(1)$ that govern the possibility of a switch to round 1 (or agreement in round 1) happening: $o_1(1) < 1 - t_2(1)$ (Player 1 wants less than what would satisfy Player 2), $o_1(1) = 1 - t_2(1)$ (Player 1 wants what satisfies Player 2), and $o_1(1) > 1 - t_2(1)$ (Player 1 wants more than what satisfies Player 2). A switch is possible in the first two, if $o_1(1) \leq 1 - o_2(2)$ is true, but not in the last. The probability of $o_1(1) = 1 - t_2(1)$ to happen however, is close to zero for real valued search. On the other hand, *it is worth keeping in mind, as will be obvious shortly, that if* $o_1(1) \geq 1 - o_2(2)$*, the only possibilities remaining are* $o_1(1) = 1 - t_2(1)$

and $o_1(1) > 1 - t_2(1)$*, hence a switch is not likely to happen.* Note that if $n > 2$, the above discussion still holds, as we only ever have to compare values from a round against those of the immediate previous round, so as to tell if a switch to the previous round may happen.

To illustrate this schematically, consider Figure 3.15. We make use of a line diagram to explain what happens to the strategies of both sides as evolution progresses. In the figure, the horizontal axis represents the round of play and the vertical axis, time. The boxes shown for time t and $t+1$ contain the strategies of both sides (offers on the table during negotiation) from the point of view of Player 1. This means that if Player 2 wanted an amount x, then the diagram will show how much x means for Player 1. In our simple example where we consider a linear relationship between the utility functions of both players, this would mean $1-x$ is being given to Player 1 by Player 2. Let us call the line joining the offers across rounds as the *strategy line*. Note that the sortedness property assures that Player 1's strategy line *slopes downwards* whereas Player 2's *slopes upwards*.

We can see that Player 1, at some point, positions its strategy line such that Player 2 rejects Player 1's offer (0.015) and offers back more (0.02) than what it just rejected in the previous round (the box at time t, where the agreement happens in Round 2). This allows Player 2 to change its strategy such that at $t+1$, it accepts the offer of Player 1 in the previous round (the box at time $t+1$, where the agreement happens in Round 1). Player 2's fitness has thus increased, which is why it changed its strategy in the first place, whereas, since the fitness is now evaluated at Round 1, Player 1's fitness for the same *old strategy* has decreased (note that a *new strategy* that may have resulted in lower fitness than before, would not have been accepted by the evolutionary algorithm). *A crucial result to note here is that sortedness alone is not enough for the co-evolutionary process to converge to optimality.*

Sortedness implies that the search space is restricted to strategies with this monotonicity property. The algorithm is not allowed to get lost with the loss in selection pressure at local sites within a strategy due to local evaluation. In so doing, we prevent the algorithm from diverging (by letting strategies switch their agreement rounds) to some extent. The less the randomness when reaching the equilibrium, the more easy it is to understand why it may not stay there. The

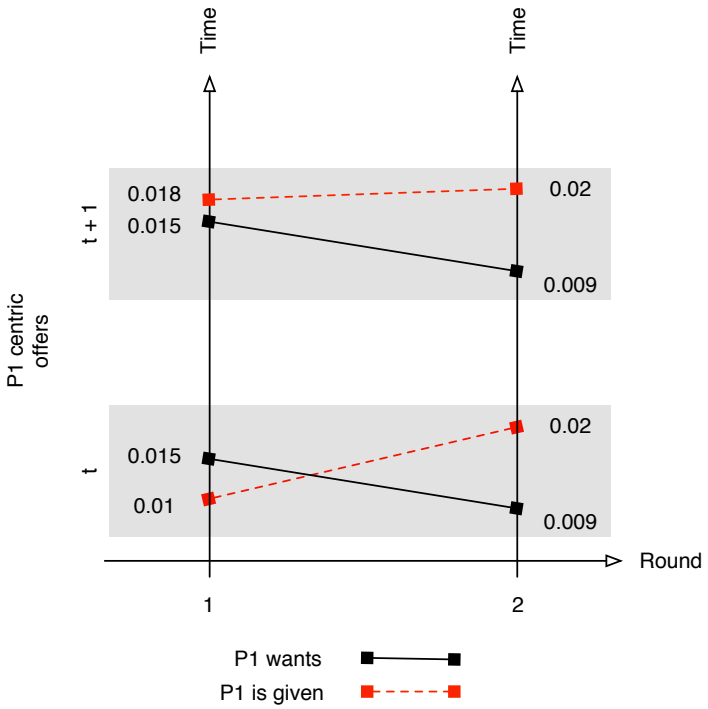
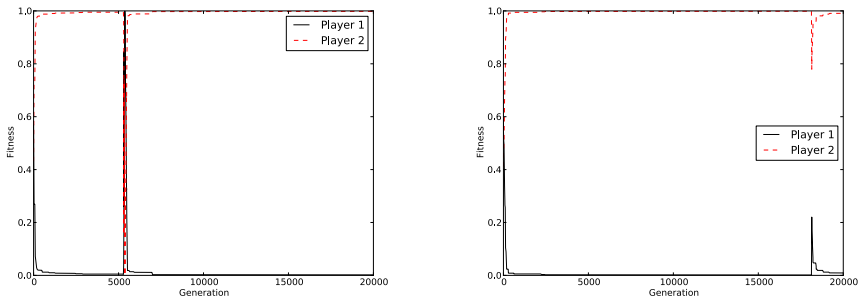


Figure 3.15: Line diagram for divergence with the 'sortedness' property in place, for $n = 2$.

line diagram (Fig 3.15) clearly shows the problem that still remains. Note that limiting the search space introduces some information about the kind of strategies that we want but the fact that the agents are boundedly rational continues to hold. They would be rational if they could construct the game tree and 'reason' using backward induction. All we have done however, is to restrict the game tree but not to the extent that only leaves the equilibrium to remain in it. There is no backward induction that the agents reason with. *The algorithm still has to search the game tree.*

Another way of introducing sortedness into evaluation is by penalising unsorted strategies before they play for evaluation, as opposed to during the game as we do, checking until the agreement round or the last round, wherever the game ends. No matter how we introduce sortedness, it does not solve the above mentioned problem as can be seen in Figure 3.16. At each evaluation step (when two individuals play a game with each other), we check if a strategy is sorted. If it is not, we assign it a negative fitness. If a player is sorted but its opponent against which it is being evaluated is not, the player gets zero fitness. The rest of the algorithm remains the same as before the rules version of Section 3.6.3.2.



(a) Diverging run 1.

(b) Diverging run 2.

Figure 3.16: Diverging runs for $n = 2$ with the 'sortedness' check in place.

In order to rectify the problem with the sortedness property alone, a third rule is introduced into the evaluation procedure. This directly tackles the problem as described in Figure 3.15.

The rule can be stated as:

Rule 3: A player's current offer's utility should be greater than or equal to the

utility rejected (utility from the rejected offer of the opponent) in the previous round.

Considering Table 3.2, if a player, as a responder, has rejected an offer from its opponent in a certain round ($t_2(1) > 1 - o_1(1)$ or $o_1(1) > 1 - t_2(1)$), then in the immediate next round, the player should not want anything less than the offer just rejected ($o_2(2) \geq 1 - o_1(1)$, which translates to $o_1(1) \geq 1 - o_2(2)$). If the player does indeed concede more than the offer it could have had in the previous round, we want the co-evolutionary solution concept to penalise this behaviour, in a manner, restricting the search space even more, allowing the algorithm to search better. For either agent, this restriction on the search space is relative to the opponent, and is suggestive of the kind of opponents it would play with to construct and move along the gradient towards equilibrium. Note that the agents have not been given any deliberative abilities, thus their bounds on rationality have not been modified in any way. The evaluation procedure that helps form a gradient over the search space is what is being modified. Again, we have restricted the game tree by making certain moves infeasible from the point of view of the evaluation procedure, and not given the strategies the ability to reason.

A consequence of introducing this rule in the evaluation procedure is that the algorithm leads to agents not being able to switch back to agreeing in any previous round, once agreeing in some round. Figure 3.17 shows the line diagram that explains this phenomenon. Imagine the agreement happening in Round 1 at time t (the lower box). Player 1 can change its strategy to the profile shown in the box for time $t+1$ and gain in fitness, since Player 2 concedes more in Round 2 (due to the sortedness rules). Note that Player 2 has not changed its strategy. The only reason why Player 2 will change its strategy is to gain in fitness. It cannot do anything in Round 1 as it is the responder there, so it gets what it is given, but it can of course change to disagree in Round 1, in which case the profile of strategies should look like in the box at time $t + 1$. Once the strategy profile is like in the box at time $t+1$, we can see that, any switch to an agreement in Round 1 will mean a lowering of fitness for Player 1, and Player 2 will have to concede more than it concedes in Round 2 (thus lowering its fitness too). This, in effect, leads to a *one way switch* towards the higher rounds for any $n > 1$. Note that this one way switch becomes effective as soon as there is an agreement

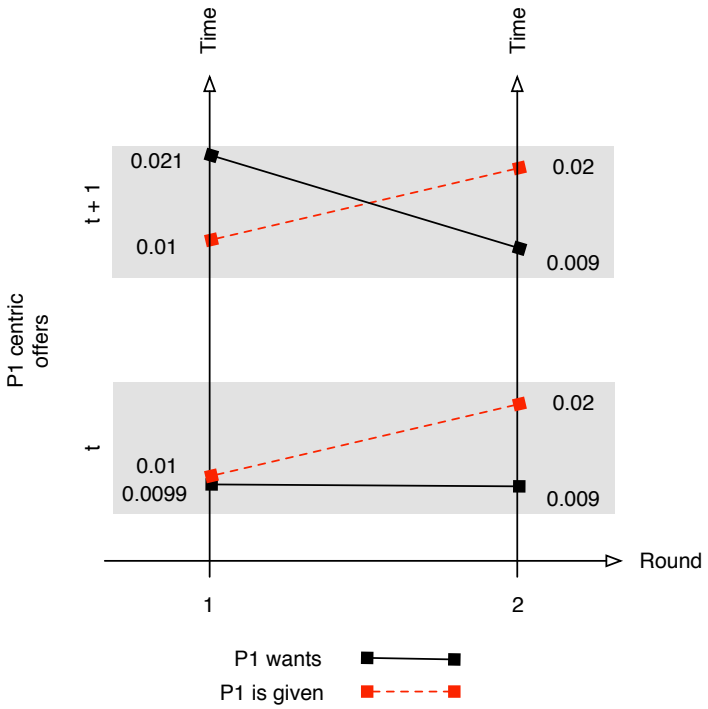


Figure 3.17: Line diagram showing a *one way switch* from low to high round, for $n = 2$.

between parental opponents adhering to the rules (both have positive fitness). Once the algorithm has lead to strategies agreeing in the last round, it is easy to see that the last responder is on a fitness plateau i.e. any threshold it uses to respond will have the same fitness, as dictated by the offer of the offerer (as long as there is an agreement, which is going to be the case). The last offerer will thus have the advantage and will continue conceding less and less until the responder reduces (randomly) its threshold to near zero, converging at infinity. In essence, the last offerer will get everything and the last responder will get nothing, which is the equilibrium outcome for a game with $n \geq 1$ i.e. any number of rounds.

Given our discussion of the solution to the problem of local evaluation, the evolution of agreement rounds is monotonic, after an agreement in some round by rules adhering parental opponents is realised, on all the runs for all $n > 1$. In order to show that this is indeed the case, we conduct a bookkeeping process during the course of evolution for all n. For each generation, we set a tag, which is initialised to n. Considering 10 runs for each n, for every n, we go through each run and check if the exit round in the previous generation was higher than the exit round in the current generation (once there is an agreement). If this is the case then we tag the current generation by subtracting 1 from its current tag value. Monotonicity deems the tags to remain the same as the initial value i.e. n, for all n. We plot the tags so obtained and it can be seen in the Figure 3.18, that monotonicity indeed holds.

Figure 3.19 shows the behaviour of the three rules version of the algorithm for various values of n (not all are shown for clarity). The trends towards convergence to equilibrium are clear. When n is odd, Player 1 evolves towards getting everything and Player 2, nothing. When n is even, Player 2, evolves towards getting everything and Player 1, nothing. As n increases, more time/evaluations are needed for the players to converge to equilibrium as can be seen in the figure. This is particularly clear for $n = 10$. We ran the algorithm for $n = 10$ for 3 million generations and do indeed see the trend towards convergence better (Figure 3.20(a)). The evolution of the rounds is monotonic, as can be seen in Figure 3.20(b).

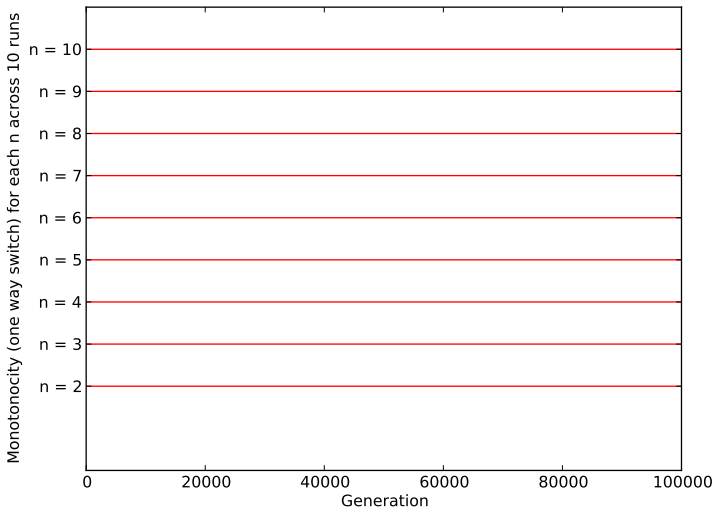


Figure 3.18: Monotonicity (*one way switch*) for each n across 10 runs.

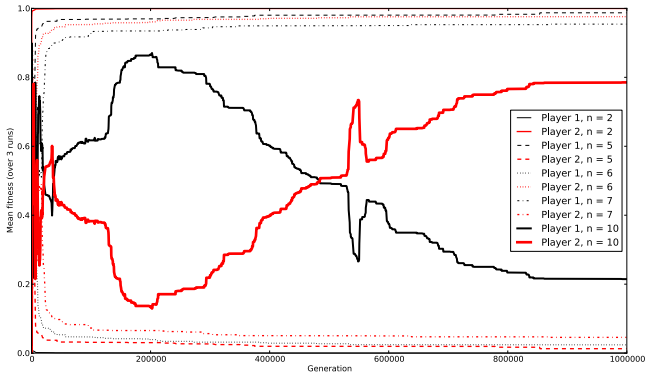
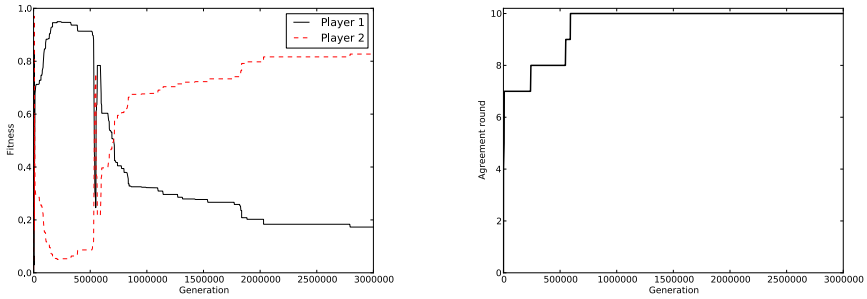


Figure 3.19: Convergence/trend towards convergence to the equilibrium with the three rules in place. Mean fitness for both players across 3 runs, for a million generations.



(a) Fitness progression. (b) Agreement round progression.

Figure 3.20: Trend towards convergence to the equilibrium with the three rules in place for $n = 10$.

3.6.3.4 Selection and Representation

Selection and representation are the remaining parts of the co-evolutionary solution concept that may be modified. But, since we have achieved convergence to the equilibrium solution for bargaining games with $n \geq 1$ already, we do not change them. In effect, we have avoided mis-interpreting bounded rationality using the notion behind co-evolutionary solution concepts, giving us insights into what it does not mean (in computational or algorithmic terms) within socio-economic simulations.

This allows us to now take bounded rationality seriously and understand how it affects co-evolution. In other words, we can now consider quantifying bounded rationality as an explicit computational or algorithmic entity to understand how it really affects co-evolution towards specific outcomes so as to leave no room for it to be mis-interpreted within simulations. In addition, since we are interested in socio-economic simulations, and reality (as verified by experimental economics research) does not conform with rational game theoretic equilibria (in reality, bounded rationality is indeed a cause or at least a reasonable assumption to be a cause for deviations from rational outcomes), an explicit consideration of bounded rationality within simulations can only help enable the cause of such simulations. If bounded rationality could be quantified using the representation of the strategy alone, and the bounds systematically varied, we could understand the relationship between the bounds and the co-evolutionary process better. Right now we

can only say that with bounded rational agents the algorithm can converge to equilibrium. To gain more insights into how bounds affect a converging algorithm, we will be considering the representation of strategies in more detail in the next Chapter.

We want to do this to show how unassessed assumptions about socio-economic phenomena within co-evolutionary simulations can further be scrutinised. This relieves us from mis-interpreting such socio-economic phenomena, giving a firm algorithmic handle on the phenomena, and thus help further engineer co-evolution for such mis-interpretations to be methodically avoided. The methodology for such a treatment of co-evolution is the subject of the next Chapter.

3.7 Generalisation to More Complex Games

For the simple game considered here, we know that previous work was unable to achieve convergence to the equilibrium solution from Section 3.5. Moreover, when the game is made more complex by extending it to having breakdowns during the bargaining process, by introducing a continuation probability $p < 1$, selection of equilibrium is still not possible. However, the interpretations of the simulations and results are still carried out in [59]. The co-evolutionary process is now interpreted as simulating selection of outcomes in games from experimental economics research, as well as theoretical equilibrium outcomes for different games altogether, both for the case where $p < 1$. It is clear that equilibrium selection in the games under question [59] was not achieved. Added to this fact, considering that equilibrium selection in simple games was also not achieved, it can only be said with little confidence, that the co-evolutionary process considered in previous work simulates equilibrium selection via socio-economic learning for bargaining games. We did not learn anything about the way to simulate socio-economic learning for equilibrium selection, nor anything about the co-evolutionary process itself in previous work. What we saw was a possible application of co-evolutionary algorithms to the problem of socio-economic learning for bargaining games, and the possible socio-economic interpretations that this application suggests. It only makes sense to understand if a method generalises to more complex games, if it does what it is meant to for simple games. This discussion is just to re-iterate the

fact that without the proper consideration of a co-evolutionary solution concept, ACE may be in danger of going off track, being mislead, and in general remaining premature as a field.

With our consideration of the co-evolutionary solution concept, we do learn how to simulate equilibrium selection. We do know how to methodically avoid mis-interpreting bounded rationality from the point of view of co-evolutionary solution concepts, but this kind of avoidance is still not satisfying. The reason for this not being satisfying is that we still do not know exactly what bounded rationality means within the simulation, i.e we cannot concretely discuss the relationship between bounded rationality and co-evolution other than knowing that it does not hinder equilibrium selection, and we want for it not to be mis-interpreted. If we do not know what it means, we cannot altogether rule out the possibility of its relationship with co-evolution being mis-interpreted. We thus want to delve deeper into the algorithm and systematically understand what bounded rationality could mean, before we try to consider how well it generalises to more complex games. We leave the latter as future work however. Note that we do understand that generalisation to more complex games may necessitate more refinements to the co-evolutionary solution concept implemented in this Chapter.

3.8 How Does Bounded Rationality Affect Co-evolution?

We now know that selecting infeasible and incompatible players can be explained by examining, understanding and modifying the variation and selection schemes in place in the co-evolutionary process. We know that local evaluation of individuals leads to a loss in selection pressure, causing strategies to change randomly, leading to divergence. Monotonicity, together with a third rule considered in Section 3.6.3.3 come to the rescue in this case. This loss of selection pressure can thus be explained by examining, understanding and modifying the evaluation scheme in place in the co-evolutionary process. As also explained throughout Section 3.6, these modifications still *preserve* the boundedly rational nature of the agents. If the agents were fully rational, by definition, we would not need a

learning mechanism in place for them to emulate rational behaviour. All we can say for now, is that we get convergence to equilibrium with boundedly rational agents. So, as can be seen, we still do not understand what bounded rationality really means for its affects on co-evolution. To consider bounded rationality as the main argument for deviations in previous work is thus just a misleading guess and hence, premature. To understand bounded rationality better, a systematic study of it, provided it can be described quantitatively, is necessary.

Knowing equilibrium selection is possible with our solution concept, one way to get a firm handle on bounded rationality is by associating bounded rationality with an element that defines the solution concept, such that it can be systematically tweaked. This can allow us to deviate systematically from our baseline bounded rationality results obtained so far. Agent strategies are a good candidate for this, since any agent's behaviour is dictated by its strategy at any given time. Bounds can be constructed by understanding the computational capabilities of the agent strategies. These computational capabilities can be quantified and tweaked to understand the relationship between bounded rationality and co-evolutionary simulations comprehensively. This is the subject of the next Chapter.

3.9 Summary

The idea behind this Chapter was to discuss the nature of ACE, as a research field, as it stands, and try to link it with co-evolutionary solution concepts, a notion of much importance in understanding co-evolutionary algorithm design. The reason for understanding this link is that co-evolutionary algorithms have largely been used off-the-shelf as tools for modelling socio-economic learning. Given their off-the-shelf usage, we deem a socio-economic interpretation of the elements of the evolutionary process and the outcomes thereof as potentially misleading.

The objective of this thesis is to lay out and apply a methodical framework, using which co-evolutionary algorithms may find a better use as simulating socio-economic learning, the idea being for socio-economic phenomena that get used in order to explain the co-evolutionary process and outcomes, may not get misinterpreted within simulations. We lay out the first part of the methodology in

this Chapter, by promoting the notion behind co-evolutionary solution concepts. We then take a particular line of work from ACE literature as a case study and apply our framework. This line of work uses co-evolution as a socio-economic learning and adaptation model to learn equilibrium selection in bargaining games. We show the misleading character of the use of these algorithms. We analyse and refine the co-evolutionary solution concept so that it does indeed result in equilibrium selection without resorting to holding a socio-economic phenomenon like bounded rationality as being responsible for the cause for any deviations encountered, until convergence results with boundedly rational agents.

Given the fact that bounded rationality generally defines the nature of agents within ACE, although it may seem *attractive* to use the term as explaining deviations from the theoretical fully rational outcomes, as is done in experimental economics and in reality, it may *not necessarily be the right thing* to do in simulations. This goes for the case study considered, as deviations from theory, although explained by bounded rationality, also deviate from experimental economics results at the same time (where bounded rationality is supposed to cause deviations too). It is thus very unclear as to what bounded rationality means in previous work and how it affects co-evolutionary simulations. The first part of our methodical framework allows us to scrutinise and challenge the usage of bounded rationality when used as an explanation within simulations in an unassessed manner, allowing us to avoid mis-interpreting it by letting us avoid explanations based on it, as we show by applying the framework to previous work. We still do not know what it concretely means within and for (i.e. its role within) the simulations however, so we cannot altogether rule out the possibility of its relationship with co-evolution being mis-interpreted. This fact further encourages us to make concrete links between ACE and co-evolution. As such, we are interested in showing how co-evolution can be made more tangible from the point of view of bounded rationality, or socio-economic phenomena of interest in general, which leads to the next part of our methodical framework, that we lay out and apply in the next Chapter.

Chapter 4

Engineering Co-evolution with Bounded Rationality

4.1 Introduction

Our agents, in the co-evolutionary solution concept engineered in Chapter 3, are boundedly rational. This can be confirmed by the fact that agreements are pushed to the last round. Full rationality entails agreements happening with no delay, which is clearly not the case. However, we do converge to the equilibrium outcomes with our boundedly rational agents. It would be premature to interpret (and indeed mis-interpret) simulations in socio-economic terms without attempting to get the simulations to achieve the desired outcomes by challenging the use of socio-economic terms as interpreting the algorithmic workings of the simulations, i.e the socio-economic terms are better not left unscrutinised. We showed how previous work used bounded rationality to prematurely interpret the simulations for their inability to achieve convergence to the subgame perfect equilibrium. We then showed how working on the co-evolutionary solution concept can let us achieve the desired outcomes, and challenge the interpretation of bounded rationality. By doing so, we show how socio-economic phenomena of interest can be challenged for their soundness within simulations, and understand whether or not they are being mis-interpreted. Interpreting simulations of reality, where bounded rationality abounds, can only be done justice to, if inter-

pretations of simulations of theory (reality being a deviation from it) are sound in the first place i.e. we know, algorithmically speaking, what the simulations are doing. We saw how the game theoretic solution concept of subgame perfect equilibrium can be implemented algorithmically. The notion behind co-evolutionary solution concepts was thus made explicit via the first part of our methodical framework, helping avoid mis-interpreting bounded rationality. Treating the elements of the implemented co-evolutionary solution concept with a clear top-down goal of converging towards the subgame perfect equilibrium, lead to the necessary refinements to these elements. This showed that bounded rationality was being min-understood in simulations in previous work [59]. That said, we cannot still rule out the possibility that bounded rationality may not get mis-interpreted, because we, for now, only understand what it does not mean within simulations. To help remove the possibility that it gets mis-interpreted, we have to ask the question of what it really means for simulations. Thus, the issues addressed in this Chapter revolve around the theme of the effect bounded rationality has on co-evolutionary simulations, which necessitates a concrete understanding of what it means within simulations. This, as we will see in this Chapter, further allows us to analyse and refine the co-evolutionary solution concept that specifies the co-evolutionary algorithm (hence, the ACE simulation) from a bottom-up socio-economic viewpoint, as opposed to subgame perfect equilibrium helping guide refinements from the top-down. Hence, this Chapter is about laying out and applying the second part of our methodical framework for analysis and refinements of co-evolutionary algorithms for the purpose of simulating socio-economic learning such that there is no room left for mis-interpreting socio-economic phenomena of interest which get used to explain the workings of these algorithms. As such the co-evolutionary process and outcomes are also prevented from being mis-interpreted in socio-economic terms.

Although we do not intend to match simulation and reality in this thesis, we propose a methodology that can be undertaken so as to enable practitioners find matches. A thorough study of co-evolutionary solution concepts for the notion's use in ACE from a viewpoint complementary to that in Chapter 3, is the main idea presented here. This thorough study is made possible by proposing the idea of what we call reconciliation variables. We engage in understanding socio-economic

interpretations of co-evolutionary simulations from a bottom-up viewpoint, by associating phenomena of interest, that may be seen as the cause for deviations from expected emergent phenomena, with the elements that define the co-evolutionary solution concept. This association is the first step towards getting a handle on the phenomena of interest from the viewpoint of co-evolutionary solution concepts. Once this is done, the elements are parametrised, thus parametrising the phenomena of interest, such that they can be systematically varied, and their effect on the co-evolutionary emergence studied. The phenomena of interest thus parameterised are the reconciliation variables. This methodology results in finding much deeper connections between the co-evolutionary solution concept implemented and the socio-economic phenomena (in our case, as a case study used throughout the thesis, *bounded rationality*) under investigation, the role of which may have been mis-interpreted in the ACE literature, from an explicit computational and quantitative point of view. Moreover, the fact that bounded rationality is generally assumed as implicit in the simulations, which in fact lead to it being mis-interpreted in previous work, as shown in Chapter 3, our investigation in this Chapter allows us to make a concrete link with simulations. This might in fact be necessary if real world socio-economic learning needs simulating using co-evolutionary algorithms, because otherwise, we it may be difficult to understand what degree (which is unknown) of bounded rationality inherently present in off-the-shelf algorithms may indeed reasonably represent the real world, thus making the choice of the algorithm for the purpose of the simulation desired limited by the algorithms available for use.

We essentially promote the idea of having a tangible computational handle on socio-economic phenomena, by way of actually designing one in the sense of overlaying reconciliation variables over co-evolutionary solution concepts. There are socio-economic phenomena which have, erstwhile, only been looked at qualitatively or even assumed to be implicit within the field of ACE research, as in our case of the phenomenon of bounded rationality, and the methodology discussed in this Chapter enables understanding the connections such phenomena may have with the co-evolutionary solution concept in a more methodical manner. This, in turn, allows for a more concrete interpretation of the co-evolutionary algorithms, which are adopted for simulations, in socio-economic terms.

By so doing, we open doors for the ACE research community for it to use the notion behind co-evolutionary solution concepts explicitly and tangibly, such that mis-interpretations of socio-economic phenomena of interest be avoided from within simulations methodically. The obvious benefits for ACE research that can be envisaged are:

- This should enable the practitioner to carry out similar and more detailed analysis of the simulation.
- This may eventually help the practitioner reconcile simulation and reality.

We first (Section 4.2) lay out the second integral part of our methodical framework, which we then apply in the remainder of the Chapter. This part of the framework allows for avoiding mis-interpretations of socio-economic phenomena of interest within simulations from the bottom-up, using the notion behind reconciliation variables that we propose and detail in this Chapter. We then look at the notion of reconciliation variables in greater detail in Section 4.3. This is followed by the explication of the methodology used via a case study, taking bounded rationality as our socio-economic phenomenon of interest that we want to avoid mis-interpretations of, from within the refined co-evolutionary simulation from Chapter 3. The manner in which bounded rationality is framed as a reconciliation variable and the co-evolutionary simulation evaluated in terms of this reconciliation variable is covered in Section 4.4. Section 4.5 exposes the questions one can imagine investigating using our methodology, in order to guide our evaluation of the simulation in terms of bounded rationality. We then carry out a detailed analysis of the manner in which bounded rationality affects the co-evolutionary simulation (Section 4.6) and how a particular definition of bounded rationality relates to other definitions (Section 4.7). Section 4.8 comments on the sensitivity of the co-evolutionary simulation with respect to variations in the intricacies of the specification of the reconciliation variable. Future research directions are covered in Section 4.9.

4.2 Framework Part 2: Avoiding Mis- interpretations From the Bottom-up

4.2.1 Description of the Framework

Carrying on with the objectives of the thesis, we now lay out the second part of the framework that we want the ACE community to follow in order for the simulations to not get mis-interpreted in socio-economic terms, i.e the socio-economic term does not get mis-interpreted in simulation. The second part of the framework, which also completes the framework, essentially allows for the avoidance of mis-interpretations of socio-economic phenomena of interest within simulations from the bottom-up, using the notion behind reconciliation variables that we propose and detail in this Chapter. These two integral parts together allow for analyses and refinements of co-evolutionary algorithms for their use as simulations of socio-economic learning, in order that the causes that lead to the socio-economic phenomena of interest being mis-interpreted within simulations be altogether removed, and thus mis-interpretations of the phenomena be avoided in a holistic sense. The following guidelines lay out this second part of the framework:

1. Associate the socio-economic phenomenon of interest with one or more elements of the co-evolutionary algorithm. This association must be based on some principles, specifically taking inspiration from socio-economic literature. If there is a definition of the socio-economic phenomenon in the literature, we suggest for the definition to be matched with the elements of the algorithm. If the elements can explicitly express the definition, we consider the phenomenon of interest associated. This is very important, because the point of the framework is to have mis-interpretations of the phenomenon avoided. A good back up for the phenomenon's definition in the socio-economic literature provides credibility to the interpretation of the phenomenon in terms of the associated elements.

2. The associated elements can now be parametrised, the point being for the phenomenon be expressed such that the flexibility of this expression be

controlled via explicit quantification of the parameters. As such, varying the parameter values varies the phenomenon of interest. We thus get a handle on the phenomenon of interest via parametrised and quantified elements. We call the phenomenon, thus associated, parametrised and quantified, as our reconciliation variable.

3. Define evaluation metrics that explicitly lay out the aspects of the co-evolutionary algorithm we want to understand in relation to the reconciliation variable.

4. Do a statistically sound analysis of the relationship between the reconciliation variable and the evaluation metrics. This allows us to confidently state the relationship, consequently helping avoid mis-interpretations of the phenomenon of interest within simulations. Since we explicitly, and indeed based on socio-economic principles, represent the phenomenon of interest as a reconciliation variable and are confident about what the reconciliation variable means in terms of how it affects the simulations, we avoid mis-interpretations of the phenomenon, from the bottom-up.

We carry on from the case study in Chapter 3, where bounded rationality was the phenomenon of interest considered as being mis-interpreted within simulations. Extending that case study within this Chapter, we utilise the above mentioned framework to show how mis-interpretations of bounded rationality can be avoided from the bottom-up within simulations, specifically the refined version of the simulation in previous work [59], from Chapter 3.

4.3 Reconciliation Variables

The object of the exercise is to delve deeper into understanding socio-economic simulations from the point of view of co-evolutionary solution concepts, as discussed in Chapter 3, such that they can be tangibly moulded into the kinds of simulations desired, within the realm of ACE, specifically in order that mis-interpretations of socio-economic phenomena of interest used to explain the simulations be altogether removed from happening. Having analysed and refined

one particular co-evolutionary simulation using the first part of our methodical framework in Chapter 3, we know how to approach the understanding of simulations from the solution concept perspective, taking the elements that define it one at a time, and refining them until we achieve the desired behaviour (in our case convergence to the equilibrium). By doing so, we were able to challenge the assumption of bounded rationality, specifically the role it does not play in the simulation's ability to achieve convergence to equilibrium outcomes. But, as mentioned before, we still do not know what role it actually plays within the simulations. We believe that being able to methodically mould the elements, specifically if the elements suggest links with socio-economic phenomena (like bounded rationality), can help reveal a way of managing socio-economic phenomena with simulations and understand their role, and indeed the relationship they have with the simulations. Knowing the role in explicit algorithmic terms removes the possibility that the phenomena get mis-interpreted, and in a manner satisfying the remaining concerns from Chapter 3, where the possibility of them being mis-interpreted was not entirely ruled out. We must then devise a way of methodically moulding the elements from the point of view of socio-economic phenomena of interest, and in turn, help engineer the simulation desired without the phenomena being interpreted incorrectly within these simulations. In line with this thought, we suggest one way of managing the elements of the co-evolutionary solution concept from the socio-economic viewpoint, by proposing the notion of *reconciliation variables*.

More specifically, looking at the element or elements of the solution concept a phenomenon of interest can be best associated with, parametrising this element(s), and analysing the simulation for its closeness to desired outcomes using this parametrisation, is the suggested idea here. The association of the phenomenon of interest and the element(s) of the solution concept that has been parametrised, provides a tangible hold on the phenomenon. *We refer to the phenomenon of interest thus parametrised as a reconciliation variable.* This parametrisation of the element(s) and the associated phenomenon of interest is a possible route towards the idea of reconciling simulation and expectation explicitly, by studying the range of possibilities that the parametrisation offers. To further illustrate with simple examples for this kind of treatment of a simulation,

we can take the viewpoint of variation, specifically mutation in co-evolution, as an element of the solution concept that may have *societal innovation* [39] as a phenomenon of interest associated with it. Similarly, crossover in co-evolution has been seen as *information exchange* [39] amongst co-evolving agents. These two variation operators are of course easy to parametrise (they already are specified by parameters) and thus provide a handle on the phenomenon of interest of innovation and information exchange respectively. But, there are phenomena of interest that may not have obvious associations, the case in point being *bounded rationality*, but studying such phenomena for how they affect simulations may be unavoidable and necessary (as we know from our discussion in Chapter 3) for the successful application of co-evolutionary algorithms as simulating socio-economic learning within ACE. This is where the idea of reconciliation variables becomes more interesting and a possible way forward into analysing and enabling the systematic moulding of these simulations towards what is desired or expected of them. From Chapter 3, we have an algorithm which tells us how to get to the equilibrium in the presence of implicitly present (within the algorithm) bounded rationality, so by systematically and explicitly quantifying/expressing and denoting bounded rationality as a *'reconciliation variable'*, we can understand how it really affects socio-economic simulations if we want to address the remaining concerns about there still being a possibility for it to be mis-interpreted within simulations.

The fact that we use the term *reconciliation variable* is simply because we want to stress the fact that we show how to *simulate deviations from equilibrium outcomes*, by expressing bounded rationality as a variable, wherein, varying bounded rationality causes deviations. This is because bounded rationality abounds in reality and socio-economic simulations of real world processes would then necessitate simulating deviations from equilibrium outcomes (or indeed variations in the outcomes based on other phenomena of interest which may need to be matched with desired real world outcomes, if need be) within games, the equilibrium outcomes only being valid in theory and not in the real world. Note that having a base line behaviour of the co-evolutionary algorithm understood, as we did in Chapter 3, by analysing and refining it towards achieving the desired base line behaviour (of boundedly rational agents converging to equilibrium solutions),

is important here and lets us figure out the true meaning of the phenomena of interest. If indeed the phenomena were mis-interpreted or treated superficially (i.e. taken for granted) previously, we come to know more about how they do not affect the co-evolutionary process, as we did in Chapter 3. This allows us to question how they might affect the process. If, on the other hand, they were treated with rigour, which was not the case as seen in Chapter 3, we get to understand what they mean in algorithmic, rather than socio-economic terms, which is going to help associate them with the algorithmic element or elements of the algorithm that represent them, and thus help understand the relationship they have with the behaviour of the algorithm explicitly in any case.

Taking the case of bounded rationality, we worked on the elements of the co-evolutionary solution concept from [59] in Chapter 3, helping us achieve equilibrium selection with boundedly rational agents in simulation, whereas previous work suggested that bounded rationality was the cause for deviations from equilibrium [59, 148, 149]. This tells us that bounded rationality was taken for granted in previous work. Moreover, this suggests a *lack of a proper understanding of bounded rationality in simulation*, and not just in previous work, but indeed, the whole of ACE devoted to using co-evolution simulating learning (since it is the general practice of taking bounded rationality for granted within simulations that lead to it being mis-interpreted in [59] in the first place), which should not be the case (at least in simulation), given bounded rationality is one of the phenomenon concerning human behaviour widely accepted in economics since [136, 138], putting more reality into economic theories. The question that immediately follows is, if bounded rationality is the cause of deviations from equilibrium in reality, then for the sake of ACE, as a field of research devoted to simulating socio-economic phenomenon, we must, at the very least, understand how bounded rationality affects co-evolutionary simulations. In so doing, we will be *a step closer* to reconciling simulation and reality, but more importantly from the point of view of the thesis, altogether avoid mis-interpreting it within simulations. The point is to not be mislead by something designed by ourselves. Bounded rationality is a phenomenon that is hard to define or specify in reality, but at least it can be specified and understood in simulation (just like there are specifications in the socio-economic literature), having designed these

simulations ourselves. Challenging and understanding assumptions in simulation is easier than in reality, so we take the stance of *'why not do it?'*. We indeed want to *increase the tangibility* of co-evolutionary simulations for them to be more useful for ACE, because it can be done as shown in the previous Chapter and as we further consider here, and further understanding the assumption of bounded rationality in simulations in explicit algorithmic terms is how we want to promote this idea in this Chapter.

In Chapter 3, we did not touch upon the representation of strategies. Here, we choose bounded rationality as our reconciliation variable. We choose representation as a *reasonable choice*, from the various parts of a co-evolutionary algorithm that can be picked, to associate bounded rationality explicitly with. We define it quantitatively, and thus concretely specify the malleability of our reconciliation variable. This choice helps us *focus* on one element which can be systematically varied, and has indeed been seen previously as an element which could model the boundedness of an agent [79, 122]. Moreover, literature in experimental economics research on memory [56], and a definition of bounded rationality from one of the proponents of the term within economics [133], suggests a seamless way for us to model bounded rationality with the type of representation considered in this thesis. We discuss these reasons in more detail in Section 4.4.1. The issue then becomes to tangibly understand how bounded rationality affects co-evolutionary simulations.

4.4 Methodology

In accordance with the second part of the methodical framework that we lay out in Section 4.2, to get a better handle on the co-evolutionary solution concept being implemented such that a deeper understanding of the phenomenon of interest (bounded rationality) be made possible, having already refined the co-evolutionary solution concept such that boundedly rational agents lead to equilibrium outcomes, the immediate steps that follow are:

- **Defining bounded rationality** in simulation, associating the notion with one or more elements of the co-evolutionary solution concept, such that

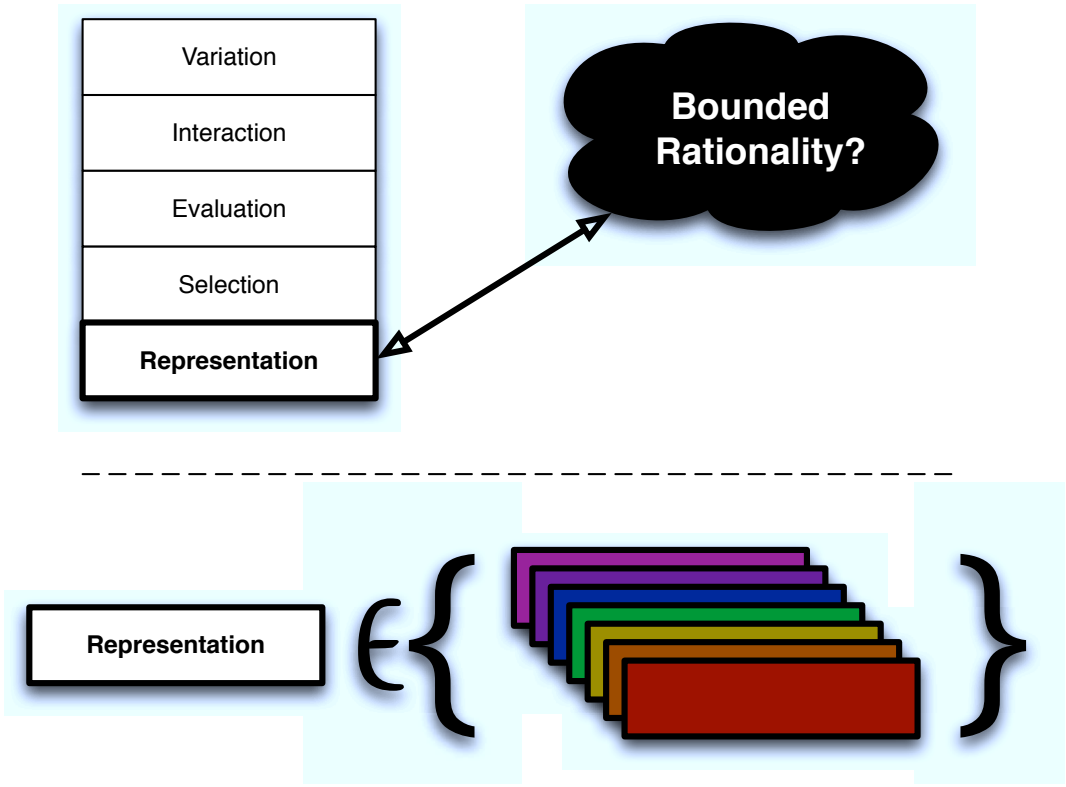


Figure 4.1: The essence of the methodology: associating the phenomenon of interest (bounded rationality) with an element of co-evolution (representation), and tangibly exploring the spectrum of bounded rationality made available by so doing.

it can be quantified and varied systematically, giving us our reconciliation variable.

- **Defining evaluation metrics** that say something about the co-evolutionary emergence, given a specific bound on rationality.
- **Establishing the relationships** between the bounds and the metrics, giving us a deeper and tangible understanding of the bounds (or phenomenon of interest) as they affect the co-evolutionary solution concept, suggesting a way to engineer the solution concept tangibly.

Figure 4.1 shows the intention of this research, *explicitly associating* bounded rationality with representation and then parametrising and quantifying the representation, giving a spectrum of bounds to explore, in order to ascertain the relationship between the bounds and the co-evolutionary emergence.

Note that the elements defining co-evolution are very much interlinked in various non-linear ways (i.e. working on one element alone may not be enough, and may take the simulation away from what is expected from it). It would then be fair to say that the phenomenon of interest may in fact be better associated with more than one element defining the co-evolutionary solution concept. However, the point here is that if one can quantify an element or a set of elements, and *reasonably* and *explicitly* associate them with the phenomenon of interest (making this phenomenon a reconciliation variable), one can then methodically figure out the role of the phenomenon within co-evolution in explicit algorithmic terms, thus not allowing the phenomenon to be mis-interpreted within simulations. One can also relate this variable to a desired emergent phenomenon, or indeed, real world emergent phenomenon (e.g. actual socio-economic learning and adaptation leading to real world outcomes), if need be, in a methodical manner. In our case, the phenomenon of interest is bounded rationality and we carry out a discussion on the way we quantify it, establishing why representation can be *reasonably chosen* as the element it can be associated with, in Section 4.4.1. This is followed by the actual *explicit quantification* of bounded rationality. The evaluation metrics that we are going to look at so as to ascertain the relationship between the bounds on rationality and the co-evolutionary process are then discussed in Section 4.4.2.

For the relationship to be empirically sound, we suggest establishing it with confidence considerations in Section 4.4.3. This is important because we now do not have a top down socio-economic phenomenon like subgame perfect equilibrium to guide algorithmic refinements. We instead have the phenomenon of bounded rationality to understand algorithmically (from the bottom up), empirical rigour thus being the way to validate its role within simulations. If we did have desired real world outcomes to match simulated outcomes with, we would still need empirical rigour of course, so as to be confident about simulating deviations as suggested by the real world outcomes, if need be. In any case, the thesis warrants designing simulations such that mis-interpretations of socio-economic phenomena be removed from within these simulations at the algorithmic level, so that we get to know what the phenomena mean within simulations. As such, we do not consider matching simulated outcomes against real world outcomes, because we do not need to for the purposes of the thesis.

4.4.1 Quantifying Bounded Rationality

We discussed the need to quantify bounded rationality in Section 4.3, once we have element(s) of the co-evolutionary solution concept associated with it. Recent work in computational and experimental economics has suggested a way to model bounded rationality.

From the computational economics side of things, the case in point is [79, 122]. As suggested in [79], bounded rationality can be considered by using the notion of *noise*, which can take the form of noisy actions or implementation of actions, noisy perception, and a combination of the two thereof. Errors in the actions implemented by the agents have motivations in Selten's notion of playing with a *trembling hand* [133], where an agent can play a strategy different from what it intended to play, with some probability. Perception errors can be seen as errors in the transmission of information [79]. Quoting Selten [133]:

> "*There cannot be any mistakes if the players are absolutely rational. Nevertheless, a satisfactory interpretation of equilibrium points in extensive games seems to require that the possibility of mistakes is not completely excluded. This can be achieved by a point of view which*

looks at complete rationality as a limiting case of incomplete rationality.

Suppose that the personal players in an extensive game Γ *with perfect recall are subject to a slight imperfection of rationality of the following kind. At every information set*[1] u *there is a small positive probability* ϵ_u *for the breakdown of rationality. Whenever rationality breaks down, every choice*[2] c *at* u *will be selected with some positive probability* q_c *which may be thought of as determined by some unspecified psychological mechanism.*"

The work in [79] considers representing agent strategies as finite state machines, and uses genetic algorithms to understand the kind of machines that may result over the course of evolution, having evolved with the specific bounds on the way they function. That work is different from ours in two respects. Firstly, the game considered therein is the iterated prisoner's dilemma or IPD game (in normal form, whereas we use sequential/extensive form bargaining games). Secondly, the evolutionary algorithm is simply picked off-the-shelf with the idea of seeing what emerges with boundedly rational agents, whereas we delve deeper down into understanding how socio-economic simulations may be analysed and refined using a methodical framework for mis-interpretations of socio-economic phenomena be avoided within these simulations. Considering the notion of boundedness in [79], the essential idea is that an agent takes an action, with a chance of this action being opposite of what the agent intended (implementation error), or an opponent's action is reported to be the opposite of what the opponent actually chose (perception error). This notion is also used in [122] to define bounded rationality, where it is shown how bounded rationality reduces the number of equilibrium outcomes in infinitely repeated normal form games. Note that there has been an independent (from ACE research on simulating socio-economic learning) study of co-evolutionary algorithms for evolving strategies and understanding the evolution towards co-operation or otherwise in the IPD game, considering noise in the implementation of actions, within co-evolutionary algorithm design research

[1] An information set for a player is a set of the possible moves from the start to the current stage in the game, made by all the players in the game, as observed by the player.

[2] Move made by the player.

[29], in order to study co-evolutionary algorithms. Noise however, was not seen as modelling bounded rationality there, further illustrating the separation of the two fields (ACE and co-evolutionary algorithm design), doing similar things, yet not feeding each other.

From the experimental economics side of things, memory limitations in terms of distortions in learning and forgetting have been looked at as possible routes to be included as dimensions for a unified treatment of bounded rationality [56]. The suggested idea is that there is a cost of *not forgetting*, which leads to boundedly rational behaviour. The greater the cost, the more bounded the behaviour. This cost is proportional to *vividness* of information, amongst other things like time elapsed since a piece of information entered memory, dynamics of the environment, whereby the more it changes, more are the costs, and the importance of information (information can be informative and useful but has a time frame within which it is useful, beyond which it becomes useless, and if this is the case, then it hampers current decision making). Considering *vividness* of information, a way of translating this idea into computation, more specifically into our co-evolutionary algorithm with the current representation, is by having randomness introduced into the behaviour of strategies.

These ideas motivate our *choice of strategy representation to be reasonable* as being the element to be associated with bounded rationality, and thus we have our reconciliation variable of bounded rationality which can indeed be tangibly modified, as discussed next. In accordance with the ideas discussed in Section 4.3 and above, we come up with the following definitions of boundedness:

- **Implementation errors** as bounded rationality: The idea here is that with a certain probability (which we call trembling probability), the action implemented or taken by an agent will be erroneous to some degree. This applies to both the offers and thresholds of an agent. In other words, when making an offer or when matching an opponent's offer with a threshold, both of which, for our case of the bargaining game with one issue are a value in $\mathbb{R}$ within the interval $[0.0, 1.0]$, the agent will have some noise added to it. Note that this noise must not take the offers outside the interval $[0.0, 1.0]$, and if it does, we simply generate the noise from the stipulated distribution again. The reason for the noise to not be outside the interval is simply

that, although the agent may be boundedly rational, it still cannot offer more than what is known to be the limit of the surplus being negotiated on. Moreover, going outside the interval would change the game (by way of increasing the strategy set available to the players), and so, the equilibrium solution for the base line bounded rationality case (i.e. equilibrium solutions considered in the previous Chapter).

- **Perception errors** as bounded rationality: The idea here is that with a certain probability (trembling probability), the current offer on the table, as put forth by an opponent, is perceived erroneously to some degree. Note that this error only applies to the offer on the table and not to the action of the opponent, nor the threshold of the agent perceiving the offer erroneously. Noise from a stipulated distribution is added to the offer on the table before being evaluated by an agent. This noise should again not take the offer outside the interval $[0.0, 1.0]$ for the aforementioned reason.

- **A combination of implementation and perception errors** as bounded rationality: In this case, not only is an action taken by an agent erroneous, but the offer on the table is also perceived erroneously. Although the probabilities (trembling probability) with which the errors of both kinds can be made, can be different, for the purposes of the thesis we keep the probabilities the same.

The above errors defining bounded rationality are specified by 2 parameters:

- **Trembling probability (T)**: This is the probability with which the agent will make an error of one of the forms mentioned above. $T \in \{0.005, 0.01, 0.015, 0.02, 0.025, 0.03, 0.035, 0.04\}$. Note that the values $\{0.005, 0.01, 0.02, 0.04\}$ were taken directly from [79].

- **Randomness in move (r)**: This is the standard deviation of zero mean Gaussian noise in the agent's action (i.e. noise in the offer/threshold) or perception of an opponent's action (i.e. perceived noise in the offer of opponent). $r \in \{0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1\}$.

The base line algorithm, which we designed in the previous Chapter, is one with agents that do not take noisy actions nor perceive opponent's actions in a noisy fashion. For these agents, $r = 0.0$. Note however, that these agents are still boundedly rational, even though they may not make errors as defined above. The solution concept that we engineered in the previous Chapter makes these agents converge to the equilibrium solution for the bargaining games considered. T and r define further degrees of boundedness, helping us understand how agents with such bounds might behave with the co-evolutionary process as specified, and compare them with the base line bound on rationality (as we may see the error free agents to possess). It can be seen why our methodical framework, promoting the notion of solutions concepts and reconciliation variables, is an important one for ACE. Using it, we are not only able to engineer co-evolution towards making boundedly rational agents converge to equilibrium outcomes, but we can further delve deeper down into and understand the notion of bounded rationality in itself (by explicitly associating it with an element of co-evolution, which in this case is the representation of the agent strategies) in a methodical manner. This allows us to further understand how it may affect agent based co-evolutionary simulations, in essence, removing the possibility for it being misinterpreted within simulations.

Essentially, with probability T, the action or perception will have a Gaussian noise with variance r^2 added to it. Mathematically, for an agent to be making implementation errors alone would result in the following offers and thresholds:

$$offer = \begin{cases} offer + N(0, r^2) & \text{with probability } T, \text{ and } N(0, r^2) \text{ resampled} \\ & \text{unless and until } offer + N(0, r^2) \in [0.0, 1.0], \\ offer & \text{otherwise.} \end{cases} \tag{4.1}$$

$$threshold = \begin{cases} threshold + N(0, r^2) & \text{with probability } T, \text{ and } N(0, r^2) \text{ resampled} \\ & \text{unless and until } threshold + N(0, r^2) \\ & \in [0.0, 1.0], \\ threshold & \text{otherwise.} \end{cases} \tag{4.2}$$

An agent to be making only perception errors would not change the offers and thresholds, but instead result in only the offer received from the opponent being erroneous, as follows:

$$received\_offer = \begin{cases} received\_offer + N(0, r^2) & \text{with probability } T, \text{ and } N(0, r^2) \\ & \text{resampled unless and until} \\ & received\_offer + N(0, r^2) \\ & \in [0.0, 1.0], \\ received\_offer & \text{otherwise.} \end{cases} \tag{4.3}$$

For an agent to be making both implementation and perception errors, the offers, thresholds and the offers received from the opponent will all be erroneous, following Equations 4.1, 4.2 and 4.3.

4.4.2 Evaluation Metrics

Given a certain degree of boundedness, as specified by a particular type of bound and setting of T and r, bounded rationality being our reconciliation variable, quantified as in Section 4.4.1, if we want to understand the nature of bounded rationality as it affects our implemented co-evolutionary solution concept, we need a way to understand its relationship with the co-evolutionary process.

This relationship can be understood by considering various aspects of the co-evolutionary process that may be affected and that are wanted to be understood, in order to engineer the solution concept better, such that it matches the practitioners needs. Establishing a relationship between the reconciliation variable and

the process is one way to mould the algorithm. At the very least, and indeed, in accordance with the second part of the framework from Section 4.2, this kind of treatment of the algorithm provides insights into the role of the phenomenon in question (that has been associated with an elements of co-evolution) within simulation, making use of the tangible handle we have on it. *It explicitly expresses the phenomenon of interest, which in our case is bounded rationality, within the simulation, for the phenomenon's role to be concretely established, so that we do not mis-interpret it.* For example, we may want to understand how bounded rationality affects equilibrium selection, a questions which we are now able to consider, since we have an explicit specification of it in terms of the parameters specifying noisy game play, overlaid on top of the agent strategy representation. Recall that such a question remained a concern at the end of the previous Chapter, since we were only able to say how bounded rationality does not affect equilibrium selection until that point. As such, the second part of our methodical framework complements the first part from the last Chapter, by allowing us to address this remaining concern.

We take the following metrics as providing insights into the workings of the co-evolutionary process, with respect to the reconciliation variable:

- **Average first hitting time:** This is the time it takes on an average (across multiple runs) for the algorithm to reach within ϵ distance (Euclidean distance) from the equilibrium solution. In our experiments, we chose ϵ to be 0.1. This reference distance can easily be any other value or considered in a different manner, specially if one wants to, say, concentrate on modelling a particular deviation from the equilibrium[1]. This metric suggests how quickly the algorithm manages to come close to the equilibrium, given a limited number of generations. Note that this is the first time the algorithm reaches within ϵ distance from the equilibrium and it may well get

[1]Instead of simply having an ϵ as a reference distance from the equilibrium, it may be more appropriate to consider a pair of reference distances, say ϵ_1 and ϵ_2, and consider deviations from the equilibrium to be measured relative to being within these two reference distances from/above/below the equilibrium. This is not required for the purposes of exposition of our methodology, hence we do not do this. One can see $\epsilon_1 = \epsilon$ and $\epsilon_2 = 0$ in what we present. However, we suggest for studies considering specific deviations from the equilibrium solution to be modelled by the co-evolutionary algorithm (in the presence of bounded rationality) to consider such a pair of reference distances. This applies to all the metrics that we discuss here.

away from it at some later generation.

- **Average length of stay:** This is the length of time in generations that the algorithm remains within ϵ distance from the equilibrium after the first hit, on an average across multiple runs. The longer the algorithm stays within ϵ distance, the less it diverges on an average, but this also depends on the maximum number of generations considered and the complexity of the game. It may take much longer for an algorithm to hit ϵ distance, with more complex games, lowering the stay given the limited number of generations. There are also times when an algorithm may hit and move away, never to have another hit, specially for more complex games, with the limited number of generations. This metric can easily be extended to lengths of stays over all the hits it gets to being within ϵ distance over the course of the evolutionary process, but we use this simplest form for the purposes of exposition of our methodology.

- **Average distance from equilibrium:** This is the Euclidean distance of the agents being away from the equilibrium solution on an average, after the first hit of being within ϵ distance from the equilibrium, across the total number of generations of evolution after the first hit, further averaged across multiple runs. This metric captures the convergence characteristics of the algorithm. If the distance is large, the algorithm generally stays away from the equilibrium, and the opposite if the distance is small. Again, if the reference distance of ϵ is varied to a value of choice for some deviation (from equilibrium) of interest, this metric may be helpful in understanding the convergence characteristics there. For the purposes of exposition of our methodology, we keep $\epsilon = 0.1$.

- **Average payoff:** This is the average fitness of the agents after the first hit, across the total number of generations since the first hit, further averaged across multiple runs. This metric also captures the convergence characteristics of the algorithm in the sense that, we have used fitness to indicate the payoff in any iteration of the co-evolutionary process, which gives an idea of the outcomes that the agents revolve around and/or towards, once close to

the equilibrium. Again, varying the reference distance ϵ (in our case fixed to 0.1), could be interesting and facilitate a more elaborate study of the solution concept being engineered, if one is to pick on specific deviations from equilibrium that one could be interested in.

- **Average Maximum/Best payoff:** Instead of taking the average fitness after the first hit, this metric tells the maximum fitness an agent achieves since the first hit, across the total number of generations since the first hit, averaged across multiple runs. In conjunction with the minimum average payoff (defined below), this metric tells the *upper limit* of the range within which the algorithm fluctuates or exhibits divergence, and as such, captures the convergence characteristics of the algorithm from this angle.

- **Average Minimum/Worst payoff:** As with the maximum average payoff, this metric instead tells the minimum fitness an agent achieves since the first hit, across the total number of generations since the first hit, averaged across multiple runs. In conjunction with the maximum average payoff, this metric tells the *lower limit* of the range within which the algorithm fluctuates or exhibits divergence, and as such, captures the convergence characteristics of the algorithm from this angle.

Table 4.1 shows the settings of the parameters for the bargaining games considered, the converging (to the equilibrium solution) co-evolutionary algorithm from Chapter 3, and the bounded rationality (which is our reconciliation variable) experimental setup.

4.4.3 Establishing Relationships with Confidence

The way we define bounded rationality, with a sound/reasonable backing in socio-economic literature, can still be subject to criticism. The same goes for the evaluation metrics one wants to consider in order to establish relationships with respect to the types of bounded rationality defined. Moreover, the bottom up socio-economic treatment of the co-evolutionary solution concept warrants rigour in the empirical analysis, for meaningful conclusions to be drawn from this analysis. We thus stress that a statistically sound analysis is in order. As such, we

Table 4.1: Parameter settings for the game, the converging co-evolutionary algorithm from Chapter 3, and the bounded rationality (as a reconciliation variable) experimental setup.

Game Parameters	Number of issues (m)	1
	Continuation probability (p)	1.0
	Maximum number of rounds (n)	10
EA Parameters	Parental population size (μ)	1
	Offspring population size (λ)	1
	Selection scheme	$(\mu + \lambda)$-ES
	Mutation model	self-adaptive
	Initial standard deviations (σ_i)	0.1
	Minimum standard deviation	0.025
Bounded Rationality Experimental Setup Parameters	Trembling probability (T)	$\in \{0.005, 0.01, 0.015, 0.02, 0.025, 0.03, 0.035, 0.04\}$
	Randomness in move (r)	$\in \{0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1\}$
	First hit distance from equilibrium (ϵ)	0.1
	Runs	50
	Generations	100,000

proposed metrics that may tell the general relationship and behaviour of the co-evolutionary solution concept with respect to the reconciliation variable (bounded rationality). We consider multiple runs of the algorithm and the metrics are all averages (across these runs). We want to be confident that these averages are indeed useful indicators of the nature of the co-evolutionary solution concept implemented. In other words, we want to be confident that we can indeed differentiate the behaviour of the algorithm given various bounds on rationality and across different types of rationality bounds. We assume that the values obtained from each of the runs of the algorithm (which then make up a sample and thus, the metric as a sample average), come from a *t-distribution*, giving us a way to establish confidence in the metrics. We calculate the standard error given by $s_m = s/\sqrt{N}$, where s is the sample standard deviation, which in turn is given by $s = \sqrt{\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{N-1}}$, μ being the sample average, and N being the number of runs (50 in our case). We then use the t value of 2.0096, given that the degree of freedom $(N-1)$ is 49. This gives us a confidence interval of $\mu \pm 2.0096 s_m$ for each value of the metric with a confidence level of 95%, telling that this value will be within this interval with a probability of 0.95, considering different samples. The value $2.0096 s_m$ is the extent on either side of the metric that the metric value will most probably lie in.

In the results that follow, we show *general trends* of the metrics with respect to bounds on rationality, as well as across types of rationality bounds. The trends shown with the metric are statistically treated for a confidence of 95% (Appendix A). We show the values of the metrics, and the values given by $2.0096 s_m$ (errors at a 95% confidence level), in the graphs in Appendix A. For the sake of appreciating the scale of the error, we plot the errors with the metrics (on the same scale as the metric) as well in Appendix A. These errors at a 95% confidence level are low (see Appendix A), and as a result, the confidence intervals are narrow. We can thus carry out useful analysis of the simulations with the metrics and indeed understand the role of bounded rationality, as it varies, within simulations, due the low errors. The metrics are useful indicators of the general characteristics of the algorithm, given the rationality bounds. As such, we can carry out an investigation of the reconciliation variable against the metrics considered. If the errors were too large, one would not be able to understand the general effect (in

terms of trends, as we do next) the phenomenon of interest has on co-evolution, given these metrics. Wherever we say significant in the discussion below, we mean statistically significant in the sense that the confidence intervals have little or no overlap. The interested reader is referred to Appendix A for a closer look at the errors at a 95% confidence shown therein for the metrics in question, given the games and rationality bounds considered.

4.5 Relationship Between Bounded Rationality and Co-evolution

There can be a multitude of questions one can investigate, given a handle on the reconciliation variable, which, by no means is an easy task, and determining what is it from the process that we are interested in observing, i.e. the evaluation metric. If one is interested in modelling deviations from the equilibrium, one way to do it is by fitting emerged/simulated deviations to real world socio-economic deviations, and explaining the emerged deviations in socio-economic terms, as has often been the case in ACE research. Although this match may well be correct, but we do not explicitly gain any insight into the algorithmic workings and nuances that lead to the match or a mis-match between simulated and real world outcomes. If ACE is to be a scientific discipline with co-evolution as the tool of choice for socio-economic learning simulations, fitting socio-economic interpretations on the algorithm is clearly not advisable, or mis-interpretations are very likely to happen. We need proper analyses and refinements of the algorithm so we can systematically narrow down onto possible simulations that may in fact model the socio-economic phenomena desired. With the explicit consideration of our socio-economic phenomenon of interest of bounded rationality as our reconciliation variable, in accordance with the second part of our methodical framework from Section 4.2, we can further this goal of avoiding it being mis-interpreted within the simulations. Understanding how it affects co-evolutionary simulations allows us to explicitly state the role it has within co-evolution, which means we get to know what it means for the simulations. This explicit understanding of bounded rationality within simulations can be realised by considering questions

like:

- Does the value of the evaluation metric change monotonically with respect to the bounds on rationality?
- Is there a threshold along the dimension (or type) of bounded rationality beyond which there are diminishing returns from the use of lesser bounded agents?
- On the contrary, it may well be that the less bounded an agent is, the more time it may take to achieve a certain level of evaluation metric, i.e. a more bounded agent may end up being more practical and in turn, real. Can this be true?
- Are there types of bounded rationality that do not affect the evaluation metric at all?
- Can we identify correlations (positive or negative) between the affects on the evaluation metrics caused by various types of bounded rationality?

We take the first three as offshoots of one question and the next two as offshoots of another. These questions being:

- How do changes in the rationality bounds affect co-evolution towards the equilibrium?
- How does a particular definition of rationality bounds (or type of bounded rationality) relate to other definitions?

This exercise allows us to look at the co-evolutionary algorithm from the viewpoint of bounded rationality. The fact that we make the relationship in explicit algorithmic terms, is to reinforce the fact that we have removed the possibility of mis-interpretations of bounded rationality within simulations using the second part of our methodical framework, and at the same time shows a way to mould simulations from the bottom-up, specifically from the point of view of bounded rationality. Note that this is being made possible to be done via our framework due to our proposal of reconciliation variables. Our framework also

helps us study bounded rationality within simulations, in greater detail, for its own sake, i.e. other definitions of bounded rationality in socio-economic literature could also be considered to be matched with the solution concept and a study similar to this exercise carried out, because our methodology does not rely on one particular definition of bounded rationality in socio-economic literature. In Sections 4.6 and 4.7, we take up these questions individually. The list of questions above is by no means exhaustive. Further investigations can be carried out and are elaborated on in Section 4.9. The graphs shown in Sections 4.6 and 4.7 depict the general trends that the bounds on rationality enforce on to the algorithm. Note that the trembling probability (T) does not play a significant role in these trends (see Appendix A for details), hence these graphs show trends averaged across T. Also, the axis value *BL* signifies *base line*, and refers to our converging algorithm from Chapter 3.

4.6 Change in Rationality Affecting Co-evolution

We want to understand how might our reconciliation variable, and thus bounded rationality, be related to the co-evolutionary process and the outcomes that result, if a particular value of the reconciliation variable (concretely specifying one element of co-evolution, which in our case for bounded rationality is the representation) and the other elements having been fixed previously (in Chapter 3), specify our solution concept.

Note that this kind of treatment of the solution concept using reconciliation variables does not necessitate the need for there being real world socio-economic situations (that need simulation) beforehand in order to validate the solution concept, but instead may help discover hidden or not so obvious situations or even synthesise new situations (possibly for the benefit of designing computational agents that may be used for computational problems via taking inspiration from economics, for example). This is however one future line of investigation and beyond the scope of this thesis.

Here, we stick to the idea of enabling tangibility to the analysis and design of the co-evolutionary solution concept such that it provides us with a way to understand the simulation better in terms of socio-economic phenomena, for it

to be confidently used as a close match to socio-economic phenomena that need modelling, i.e. help avoid the phenomena being mis-interpreted. We believe that enabling tangibility so that one can juggle with the algorithm in concrete socio-economic terms (terms, like bounded rationality, which can easily be mis-interpreted within simulations due to the ad-hoc application of co-evolution for simulating socio-economic learning) is something missing from co-evolutionary algorithm design when channeled towards agent based socio-economic learning simulations. This thesis, via the description and application of our methodical framework in the previous and this Chapter, emphasises the need for such tangibility.

As mentioned in Section 4.4, the metrics we use to evaluate the algorithm, say something about the process, that may be useful in moulding the implemented co-evolutionary solution concept towards the simulation desired, and more importantly for the purposes of this thesis, allow us to establish a concrete algorithmic link between bounded rationality and co-evolution, telling us to know what it means within the simulations. We take these metrics one at a time in this section.

4.6.1 Average First Hitting Time

4.6.1.1 Implementation Errors

We want to understand how quickly the algorithm can reach the equilibrium and what role does bounded rationality, defined as implementation errors, play in this. Figure 4.2 tells us that, across all n (number of rounds in the game), the first hitting time is low for agents with higher bounds on rationality. That is, the larger the implementation errors made by an agent, the quicker it will have a first hit within ϵ distance from the equilibrium on an average. Moreover, as n increases, or the game becomes more complex, a quicker achievement of reaching within ϵ distance from the equilibrium is only possible with increasing the bounds on rationality such that, to get the same level of the metric as for small n, bounds on rationality will have to be increased. As n increases, the change in the time to hit increases less severely (and remains lower) for more bounded agents than for the ones which are more rational. For larger n this results in a sigmoidal shape

of the curve (as is evident from Figure 4.2) across r, for the metric. We also note that the trembling probability does little by way of affecting the evaluation metric. The predominant affect is due to the randomness in the actions of the players, but of course, randomness in the actions are of no use if T is zero, which is going to be the same as our base line algorithm (denoted by BL in Figure 4.2).

Since we do not have infinite time at our disposal in the real world, a realistic shortage in time available to simulate reaching equilibrium, may allow for this metric helping the design of our implemented solution concept. In other words, if the game becomes more complex, we know that it will take longer for less bounded agents, bounded in the implementation error sense, to hit the equilibrium, and looking at the first hitting time may allow a practitioner to make necessary trade-offs. On the other hand, this metric alone is not enough to understand these trade-offs, and a proper understanding of the simulation necessitates studying other aspects of the algorithm, which we are going to consider in this thesis. Taking only this metric, one can be mislead into believing that we can reach equilibrium quicker with more bounded agents, but this is of course at the cost of not converging, as will be evident with the consideration of other metrics.

4.6.1.2 Perception Errors

Figure 4.3 shows the average first hitting time to being within ϵ distance from the equilibrium for agents which are boundedly rational in the form of making perception errors. We can see that co-evolution with such agents shows different characteristics with respect to r as with agents which only make implementation errors (Figure 4.2). This suggests that defining bounded rationality in different ways can result in differences in the outcomes from co-evolutionary simulations, even when the differences are in the way noise is injected into the offers and thresholds of agent strategies, suggesting the importance of studying the assumption of bounded rationality in more detail in simulation for the purposes of ACE research. What we see from Figure 4.3 is that an increase in boundedness decreases the first hitting time, though a similar decrease happens with greater bounds on rationality, as compared to the implementation error case, for more complex games. Also, a decrease in the complexity of the game (i.e. as n de-

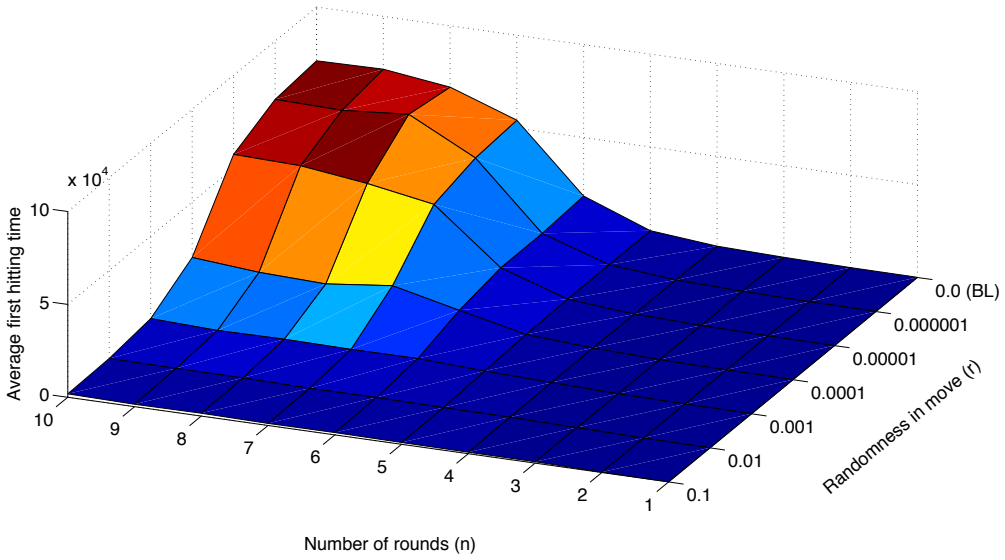


Figure 4.2: Average first hitting time, across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Appendix A.

creases) generally decrease the first hitting time, with a significant dip at $n = 7$ (across r).

4.6.1.3 Implementation and Perception Errors

Figure 4.4 shows the average first hitting time to being within ϵ distance from the equilibrium for agents which are boundedly rational in the form of making both implementation and perception errors at the same time. We can see that co-evolution with such agents shows similar characteristics with respect to this metric, as with agents which only make implementation errors (Figure 4.2), but differs from agents only making perception errors (Figure 4.3). We discuss these differences in Section 4.7. What is clear is that an increase in boundedness causes a decrease in the first hitting time. Similarly a decrease in the complexity of the game (i.e. as n decreases) generally decreases the first hitting time, with a significant dip at $n = 7$ (across r).

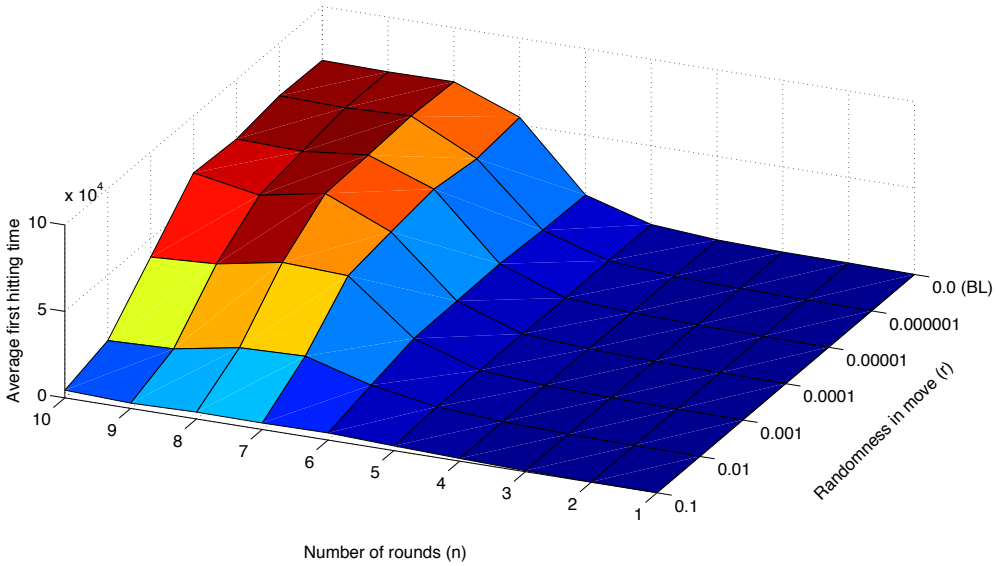


Figure 4.3: Average first hitting time, across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

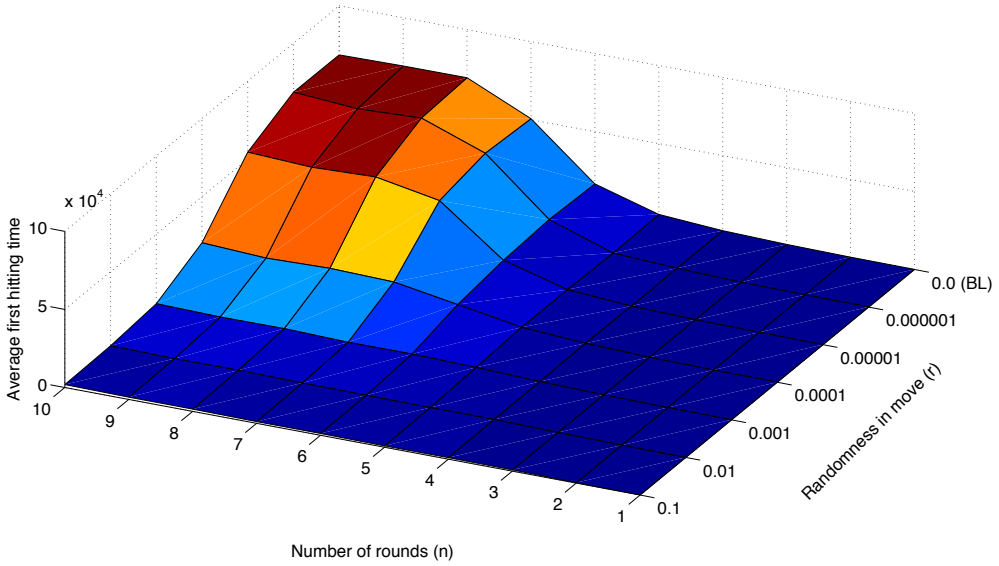


Figure 4.4: Average first hitting time, across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

4.6.2 Average Length of Stay

4.6.2.1 Implementation Errors

As n increases, the stay in equilibrium (or within ϵ distance from the equilibrium) lessens for less boundedly rational agents and this becomes similar to the stay of more bounded agents. In other words, as the game becomes more complex, a less boundedly rational agent is as good as a more boundedly rational agent when it comes to convergence within a stipulated time frame. We know from the previous Chapter that our base line algorithm converges to the equilibrium. So, given more time, the surface depicted by Figure 4.5, will look different as the difference between the base line side of the surface for small n and large n will get smaller.

Curiously, the stay increases with an increase in n for agents, specifically with bounds given by $0.000001 \leq r \leq 0.00001$, and then decreases with a further increase in n. As randomness increases, for smaller n, though it may be easier to reach within ϵ distance from the equilibrium (Figure 4.2), it may also be easier to get away from being within this distance, which, with an increase in n is harder. The three rules in place, which are part of the evaluation procedure of the agents, make it harder for a feasible strategy within ϵ distance from the equilibrium to be taken over by another strategy because of randomness in its moves, than when $n = 1$, where these rules do not apply. The stay is longer in harder games, until the game becomes even harder when it takes much longer to reach within ϵ distance from the equilibrium (Figure 4.2). So, in the limited time, the stay is lower. Increasing randomness further makes the stay short although the first hit (Figure 4.2) takes place quicker.

It can also be seen from Figure 4.5, that agents bounded with $r \geq 0.001$ remain invariant to changes in n. Comparing this with Figure 4.2 suggests that more bounded agents essentially reach and get out of the equilibrium quick, suggesting a threshold in r that makes agents behave similarly.

In general, there is a decrease in the length of stay with increasing bounds on rationality and this decrease becomes less severe as the game becomes more complex. This can be seen from the exponential turning into sigmoidal shape of the surface in Figure 4.5 as n increases, going flat towards larger n.

Note that if a simulation does not reach within ϵ distance from the equilibrium in the stipulated time, and may converge given longer time, the stay is zero, which is frequent in case of large n, hence the low value for this metric.

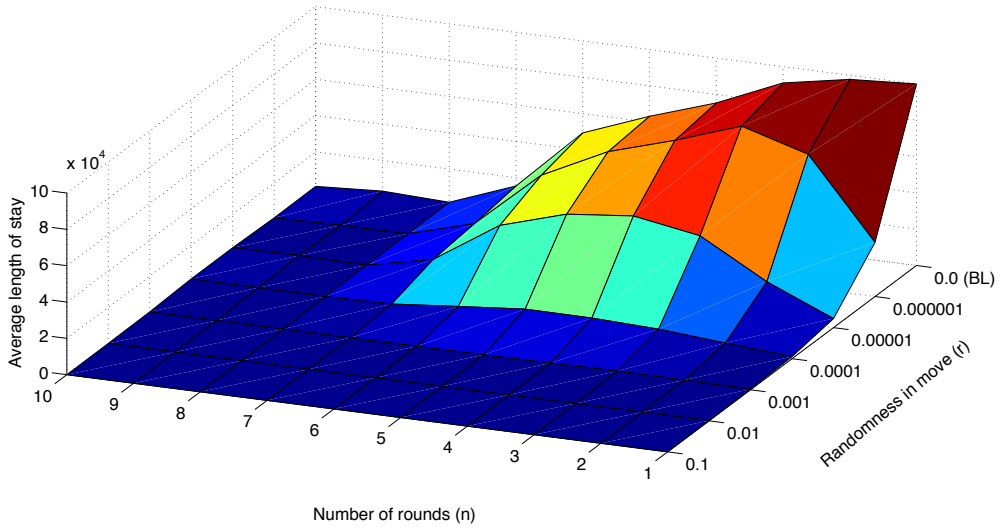


Figure 4.5: Average length of stay in equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Appendix A.

4.6.2.2 Perception Errors

The stay within ϵ distance from the equilibrium after the first hit is longer with perception errors defining bounded rationality, as per Figure 4.6, than with just implementation errors, specifically for lower n and with $r = 0.0001$. The effects on the stay are slightly different in this sense as compared to the implementation error alone, but the same trend across n and r as with the implementation errors alone, apply here.

4.6.2.3 Implementation and Perception Errors

We again see that having agents with both implementation and perception errors is similar to having agents with just the implementation errors alone defining their

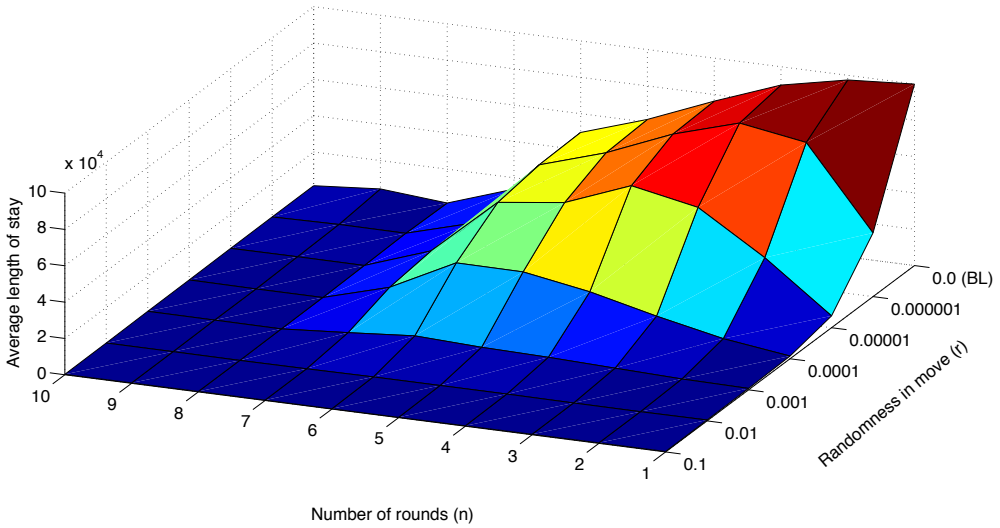


Figure 4.6: Average length of stay in equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

boundedness in rationality, for this metric too. Figure 4.7 is similar in characteristics to Figure 4.5. The effects on the stay are similar as with the implementation error alone, and the same trend across n and r as with the implementation errors alone, apply here. As such, the stay is shorter for low n and $r = 0.0001$ when compared with perception errors alone. We discuss these differences between the three types of bounds with regard to this metric in Section 4.7.

4.6.3 Average Distance from Equilibrium

4.6.3.1 Implementation Errors

The average distance from the equilibrium suggests how far the algorithm, given an agent with a certain bound of a certain kind, remains from the equilibrium over time, after the first hit within ϵ distance from it. We can take this metric as suggesting the convergent nature of the algorithm. The algorithm may fluctuate after hitting within ϵ distance from the equilibrium and greater fluctuation will show up as a large value for this metric. A greater distance but less fluctuation will also receive a high value for the metric. But, as we know, we should use this

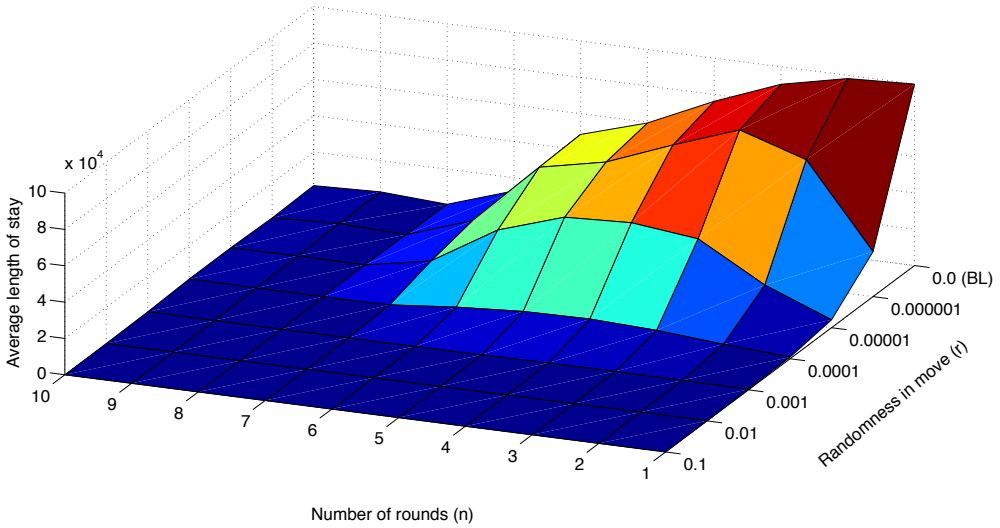


Figure 4.7: Average length of stay in equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

metric in conjunction with other metrics in order to understand how a certain bound on rationality affects the co-evolutionary process and our implemented solution concept. From Figure 4.8 we can see that the distance from equilibrium decreases with an increase in rationality when the bounds are defined as implementation errors. Thus, as n increases, a closer match to the equilibrium can only be achieved by reducing the bounds on rationality. We note that the metric indeed has a high value with more bounded agents because the algorithm diverges or fluctuates more with increasing randomness. The distance also increases for less bounded agents with an increase in n because the algorithm still fluctuates, although less than for more bounded agents, in addition to staying further away from the equilibrium as compared to more bounded agents (which may reach closer yet fluctuate more).

Interestingly, as can be seen in Figure 4.8 the decrease in distance with decreasing bounds follows a sigmoidal trend, with the inflection point decreasing along r, as n increases. Moreover, with small n (≤ 5), and agents with $r \leq 0.0001$, the metric is largely invariant to changes in n and r, whereas with larger n (≥ 6), and agents with $r \geq 0.001$, the metric is largely invariant to changes in n and r.

Since we know the base line case converges but takes longer, we can say that it needs more time to shorten the distance with increasing n. As the game becomes more complex, the amount of computation required to shorten the distance to equilibrium increases (length of stay is less with more rational agents and the first hit takes a lot longer, often not hitting at all, thus not staying either as suggested by Figures 4.2 and 4.5), so in some sense, having lesser bounds still suffers computational limits (or bounds). This suggests some links with the CIDER theory [147], with a difference that our analysis only deepens it. The theory suggests that computational intelligence (the algorithm per se) models effective bounded rationality (much the same as ACE research devoted to simulating socio-economic learning assumes for bounded rationality to be present in the algorithms from the outset), but what we have is a computational intelligence approach with a handle on the elements of the approach in terms of a concrete specification of bounded rationality for the purpose of equilibrium selection in the bargaining games considered. Saying computational intelligence gives effective rationality is one thing but proposing the idea and case of *tangibly moulding* the computational intelligence approach, challenges the assumption of bounded rationality and allows for it to not be mis-interpreted within simulations, i.e. is a step further in concretely specifying what bounded rationality may mean within simulations. Again, looking at Figure 4.8, the curve at the base line with increasing n suggests that the distance increases exponentially with game complexity (n). This is very interesting and is hard to capture with the other metrics we have covered so far viz. first hitting time and length of stay within ϵ distance, essentially because distance takes the outcomes of strategies into account rather than a computational term (viz. generations, as with the previous two metrics). Given a limited number of generations in the first place, distance provides a computational viewpoint by proxy to suggest the complexity of the algorithm in handling the game. Of course, one can increase the number of generations till one gets convergence and minimal distance with respect to the equilibrium but with increasing n, one will have to further increase the number of generations, and that may be impractical.

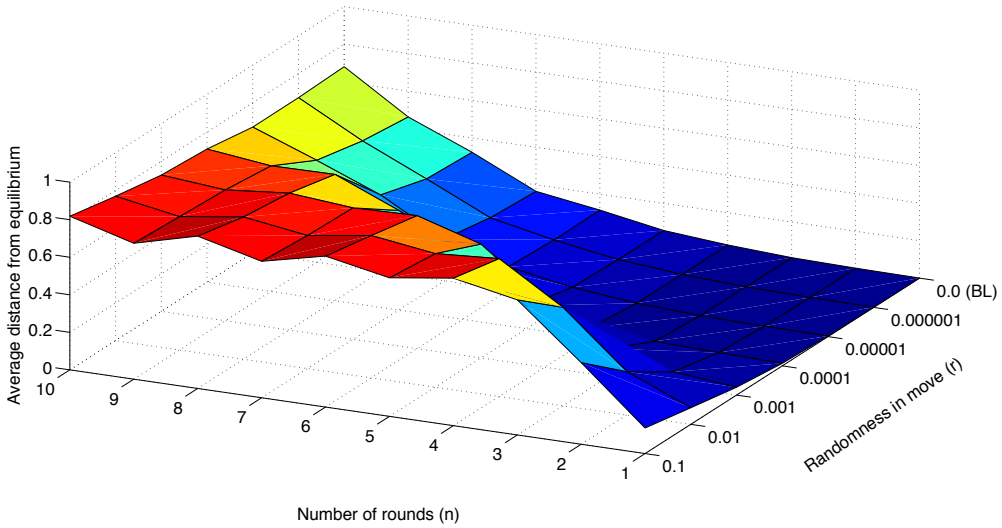


Figure 4.8: Average distance from the equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Appendix A.

4.6.3.2 Perception Errors

The average distance from the equilibrium after the first hit with agents with perception errors alone can be seen in Figure 4.9. The distance increases with the increase in complexity of the game. It also increases with an increase in boundedness. The affect on the metric that the complexity of the game has diminishes with an increase in boundedness, hence the shape of the surface which changes from an exponential rise in the distance at the base line side of boundedness to being sigmoidal with increasing boundedness. This effect is much more prominent with having agent with implementation errors alone (Figure 4.8).

4.6.3.3 Implementation and Perception Errors

As with the first hitting time and length of stay in the equilibrium, the average distance from the equilibrium after the first hit with agents with both implementation and perception errors follows a similar trend across n and r (Figure 4.10), as agents with just implementation errors defining their bounds.

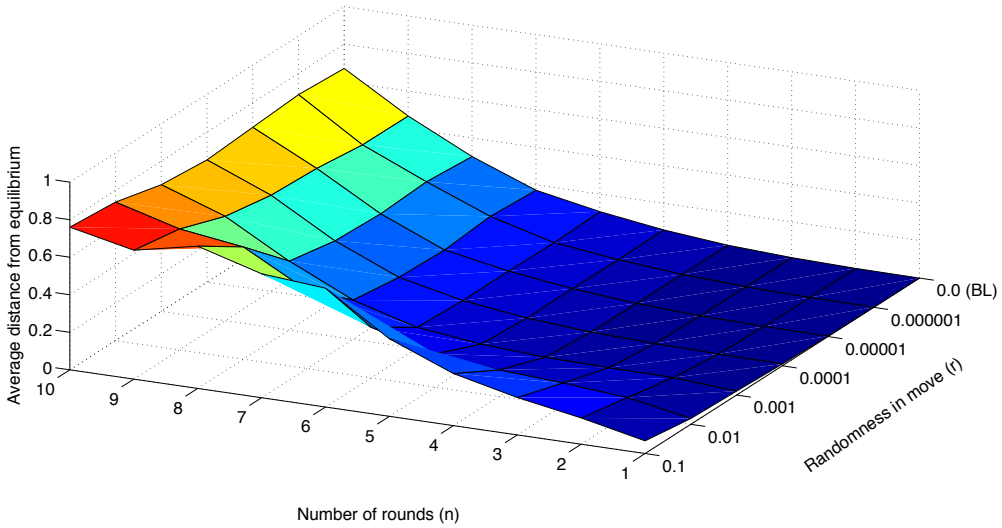


Figure 4.9: Average distance from the equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

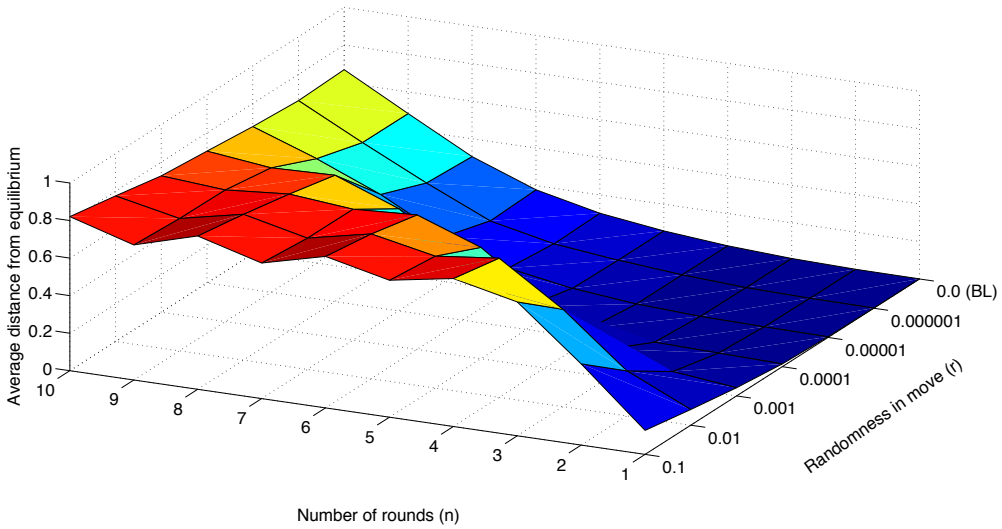


Figure 4.10: Average distance from the equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

4.6.4 Average Payoff

4.6.4.1 Implementation Errors

From the start, we have considered the fitness of the agent to indicate its payoff in any iteration of the co-evolutionary process. Note that we modified the fitness function, as part of the exercise of modifying the implemented co-evolutionary solution concept in the previous Chapter, to assign negative values to agents which do not adhere to the three rules discussed therein. This however does not change the game. As discussed in the previous Chapter, this change in the evaluation procedure forbids the algorithm from searching certain parts of the game tree, and in so doing, relieves the agents from some computational effort. We still want the fitness over time to evolve towards the desired outcomes which, in our case, are equilibrium outcomes or deviations thereof.

The average fitness after the first hit to being within ϵ distance from the equilibrium gives an idea of the outcomes that the algorithm revolves around and evolves towards, once it has come close to the equilibrium. Varying ϵ could be an interesting and a more systematic study in order to pick on the deviations from equilibria one may be interested in, but we leave that study for future work.

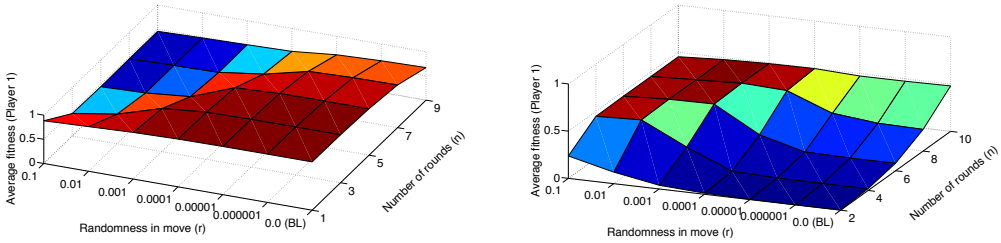
According to Figure 4.11, which shows the average fitness of Player 1 across generations since first hit, across 50 runs, for various bounds on the rationality of an agent, we can see that when n is odd, the more bounded agents get less than less bounded agents, eventually being caught up by the less bounded agents as n increases. When n is even, the more bounded agents get more than less bounded agents, eventually being caught up by the less bounded agents as n increases. The equilibrium outcome, when n is odd, is for Player 1 (being the last offerer) to receive everything (and Player 2 nothing), and when n is even, for Player 1 (being the last responder) to receive nothing (and Player 2 everything). We can see that as the bounds increase, the deviation from this equilibrium outcome increases. Figure 4.12 shows the average fitness of Player 2, and as expected, with n odd, the more bounded agents get more than the less bounded agents, eventually being caught up by the less bounded agents as n increases. And, with n even, the more bounded agents get less than the less bounded agents, eventually being caught up by the less bounded agents as n increases. Being caught up by

less bounded agents is just a visual observation and given more time (generations) the less bounded agents (specifically for $r = 0.0$) will have their average fitness more closer to the equilibrium payoff mentioned above.

In a narrow sense of just this metric, we may say that on an average, *greater bounded rationality* leads to more *fairer/equitable/symmetric outcomes*, but that may be misleading. We say this to stress the point that when considering algorithms like co-evolution for socio-economic simulation, the simulations should not be considered justified without the proper definition and consideration of evaluation metrics. If one is interested in trying to model realistic behaviour (e.g. fair outcomes), considering only the average payoff does not give a complete view of the algorithm. Other metrics need proper definition and consideration, and the algorithm evaluated against those for a better match with the required socio-economic situation. A case in point is previous work [59], where, although the algorithm seemingly results in equilibrium selection (an extension of which also suggests simulating fair outcomes) on an average across multiple runs, it wildly fluctuates (i.e. does not converge, as we saw in Chapter 3), but convergence is not looked at in conjunction with payoffs. This is of course, in addition to the proper definition of the implemented solution concept, as described by the first part of our methodical framework in the previous Chapter. As a word of caution, if we want convergence to real world outcomes where bounded rationality were to result in such outcomes, it may have to be specified in a different way, possibly with a different strategy representation (which may induce changes in other elements of the co-evolutionary process), than it is in this thesis. But this does not make our work irrelevant, essentially because we are interested in showing a methodology to analyse and design simulations, by specifically challenging the generally unassessed (hence, mis-interpreted) assumption (within simulations) of bounded rationality, and understanding what it really means for the simulations. This methodology can indeed be used with other socio-economic phenomena of interest for them to not get mis-interpreted within simulations.

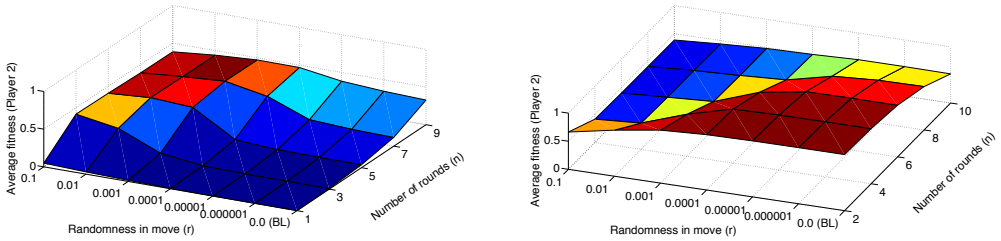
Moreover, it is possible that changing our negative fitness assignments to much smaller values than considered in Chapter 3 may bias the results. However, fitness assignment is part of the whole computational method and tangibly designing it or engineering it to achieve what we want is the object of the exercise. The

way we carry out the modifications to the implement co-evolutionary solution concept, element by element, is one way to engineer the process. A better way can no doubt replace this engineering process. The point here is to go systematic by considering elements and reconciliation variables, giving us a tangible handle on the process.



(a) n odd. (b) n even.

Figure 4.11: Average fitness of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Appendix A.



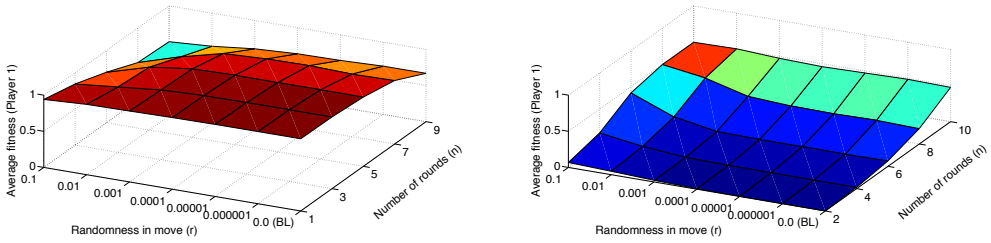
(a) n odd. (b) n even.

Figure 4.12: Average fitness of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Appendix A.

4.6.4.2 Perception Errors

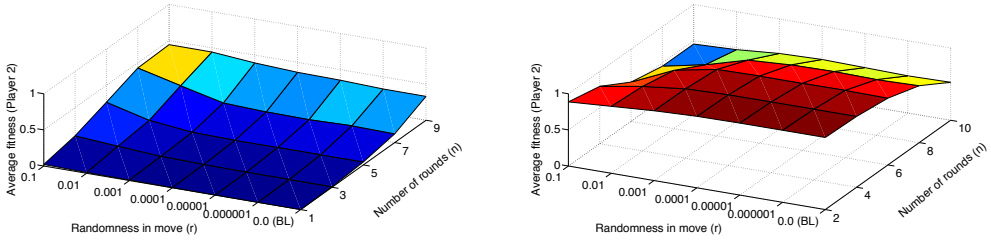
From Figures 4.13 and 4.14, it is evident that for odd n, where Player 1 should get everything in equilibrium (and Player 2 nothing), with increasing bounds on rationality, specially for greater values of n or more complex games, the agent deviates further from the equilibrium outcomes on an average. Player 2 suffers

the same deviations lending an average payoff higher than the equilibrium with increasing bounds on rationality and increasing n. In the case where n is even, the payoffs and deviations are reversed for both players as can be seen in Figures 4.13 and 4.14. The deviations with increasing bounds on rationality are smaller for agents with perception errors than they are for agents with implementation errors (specifically for $3 \leq n \leq 8$ and $0.0001 \leq r \leq 0.1$), as can be seen comparing the surfaces shown in Figures 4.11, 4.12, 4.13 and 4.14.



(a) n odd. (b) n even.

Figure 4.13: Average fitness of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Appendix A.



(a) n odd. (b) n even.

Figure 4.14: Average fitness of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

4.6.4.3 Implementation and Perception Errors

As with the previous metrics, considering agents with both implementation and perceptions errors shows similar characteristics for this metric across n, r and T, as for agents with implementation errors alone (see Appendix A).

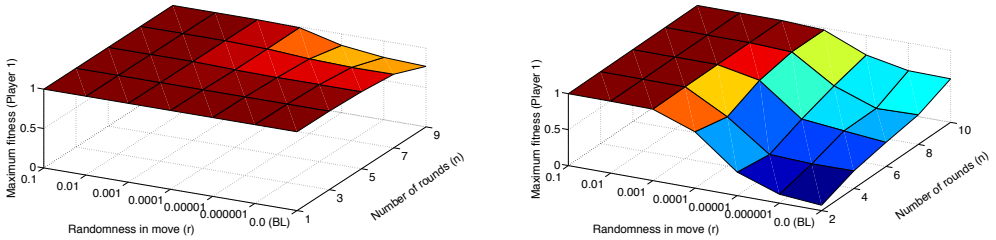
4.6.5 Average Maximum and Average Minimum Payoff

4.6.5.1 Implementation Errors

Considering Figures 4.15 and 4.16, which depict the average (across runs) maximum fitness obtained by an algorithm after the first hit to being within ϵ distance from the equilibrium and with the agents bounded with a certain value of r, there is generally an increase with increase in bounds. Beyond $r \geq 0.001$ the algorithm is essentially fluctuating in full swing such that even for n being even, Player 1 can get everything (and for odd n, Player 2 can get everything) though its equilibrium outcome is zero (as is for Player 2 when n is odd). With an increase in n, the affected algorithms are the ones with bounds $r \leq 0.001$. The average maximum fitness tends to move away from the equilibrium fitness with increasing n in general. Across n, the Figures 4.15, 4.16, 4.17 and 4.18 suggest that the range within which the algorithm fluctuates are similar towards more boundedness, becoming different with lesser bounds i.e. n plays a role with respect to these metrics for more rational agents. Lowering the bounds suggests a smaller range within which the algorithm fluctuates. Figures 4.17 and 4.18 also suggest that the average minimum fitness decreases with an increase in bounds and goes negative. This tells us that the chances of infeasible solutions being selected are higher with greater bounds on the rationality of agents. Essentially, selected strategies for agents making *different errors* (given the type of bounded rationality considered in this thesis) the *next time they play* may cause players to become infeasible if they violate the rules in place (see Chapter 3), and the chances of this happening are higher with more bounded agents. For $n = 1$, note that the rules do not apply (i.e. the minimum will hence be non-negative), and so the algorithm does not suffer from players becoming infeasible due to violations of the rules, hence the fitnesses are generally higher as compared to other n. However, the players can still disagree by making different errors than they would have, if selected, which is the main cause of fluctuations for $n = 1$ here.

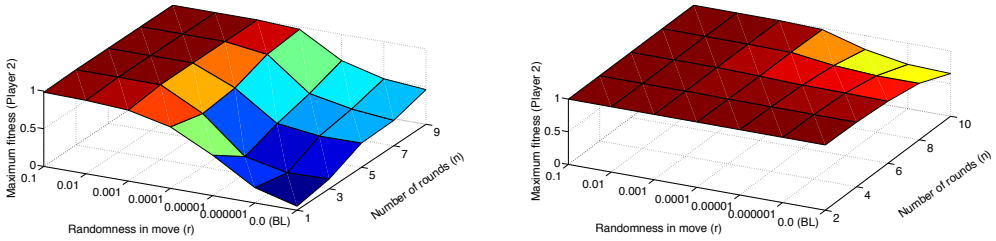
4.6.5.2 Perception Errors

The effect of perception errors on the maximum and minimum average fitness of the agents (Figures 4.19, 4.20, 4.21 and 4.22) are similar across the spectrum



(a) n odd. (b) n even.

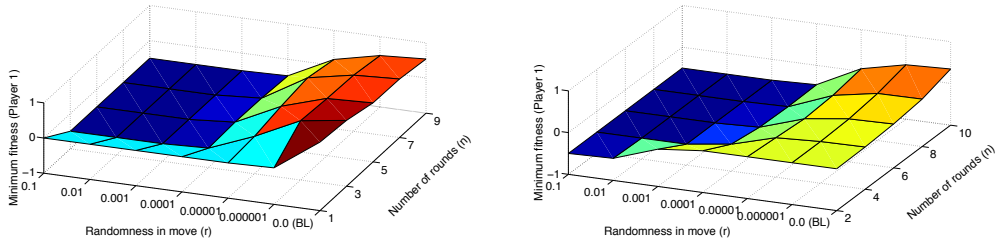
Figure 4.15: Average maximum fitness of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Appendix A.



(a) n odd. (b) n even.

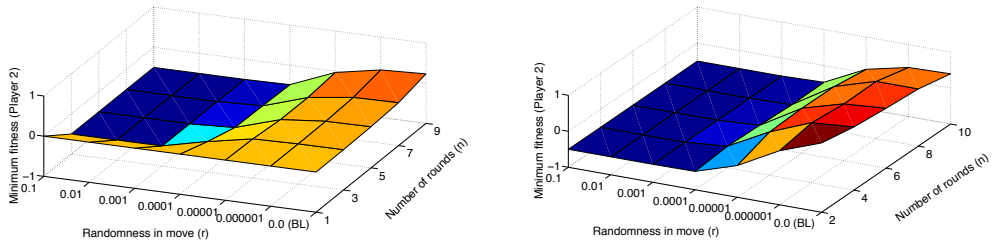
Figure 4.16: Average maximum fitness of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Appendix A.

of boundedness, however less prominent, when compared with agents with implementation errors alone (Figures 4.15, 4.16, 4.17 and 4.18). It is of interest that the minimum fitness goes negative with perception errors even though the rules that we put in place in Chapter 3 would normally apply if the strategies that agents play with, when checked, violate them, which is very likely if the implemented actions possess random noise (implementation errors). With only perception errors, the agent actions do not change. One cause of fluctuations in the co-evolutionary process are the disagreements in bargaining that may happen because of perceiving the offer of the opponent differently, thus making a good strategy seem bad. This follows the selection of strategies that indeed violate the rules, hence a negative fitness. Moreover, if the round in which the agreement



(a) n odd. (b) n even.

Figure 4.17: Average minimum fitness of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Appendix A.



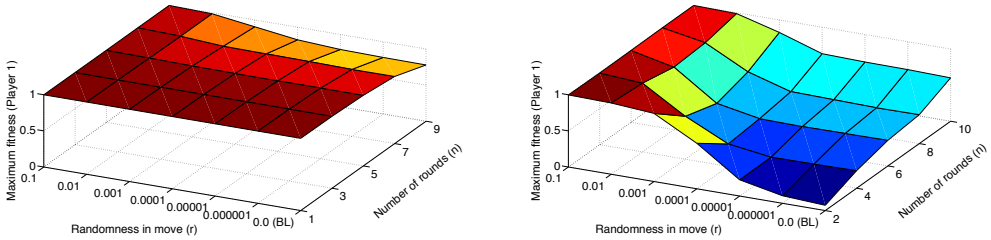
(a) n odd. (b) n even.

Figure 4.18: Average minimum fitness of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Appendix A.

for the agents was previously happening changes (due to a disagreement in that round) and they need to *play another round* (within the limit of n), perfectly fine agents, which were not violating the rules, now can violate the rules in these new rounds. This leads to a negative fitness too.

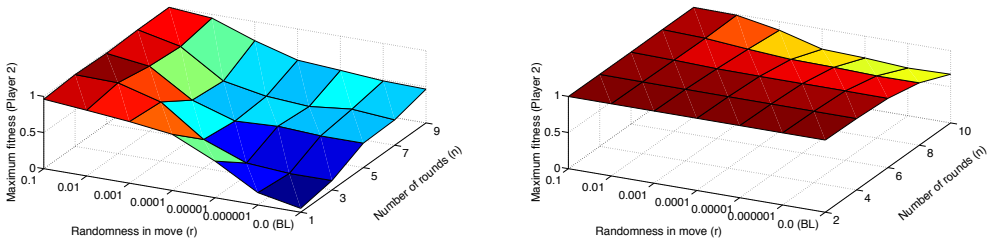
4.6.5.3 Implementation and Perception Errors

As with the previous metrics, considering agents with both implementation and perceptions errors shows similar characteristics for this metric across n, r and T, as for agents with implementation errors alone (see Appendix A).



(a) n odd. (b) n even.

Figure 4.19: Average maximum fitness of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

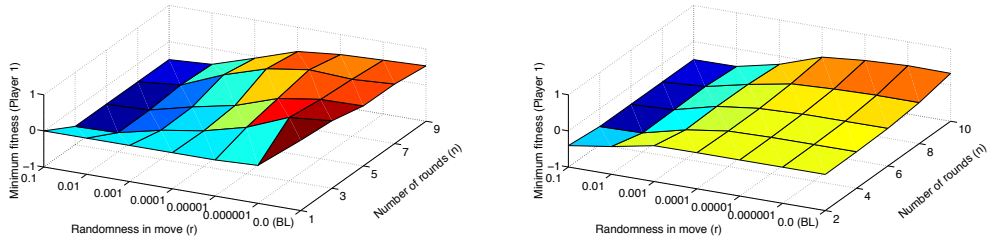


(a) n odd. (b) n even.

Figure 4.20: Average maximum fitness of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

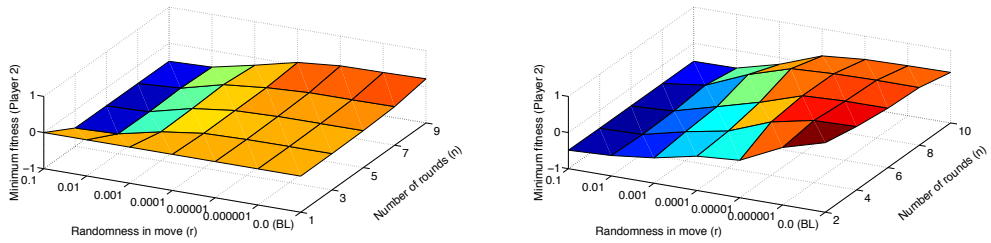
4.7 Relationship Between Types of Bounded Rationality

Considering different types of rationality, and comparing the effect they have on the evaluation metric being looked at can shed light on the relationships between the types of bounds. We take three types of rationality in the previous Section where one is a combination of the other two. Simply doing this can help deepen our understanding of the impact that a certain kind of boundedness may have as compared to other kinds, deepening our understanding of the assumption of bounded rationality that ACE makes. This further provides insights into what it could mean in simulation, allowing us to further challenging and explicitly



(a) n odd. (b) n even.

Figure 4.21: Average minimum fitness of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Appendix A.



(a) n odd. (b) n even.

Figure 4.22: Average minimum fitness of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Appendix A.

expressing it within simulations.

Taking just the case of the average first hitting time and comparing the three types of bounds (Figures 4.2, 4.3 and 4.4), we can see that adding implementation errors to the agents with perception errors results in similar characteristics as for the case of agents with implementation errors alone. A closer look suggests that the kind of bounded rationality (essentially injecting randomness) considered, results in a bound specifying perception error type of (as compared to the same bound for the implementation errors) injection of randomness still causing *significant deviations from the base line* algorithm, for more bounded agents. If one looks at the definition of these errors, as laid out in Section 4.4.1, one can see that both implementation errors and perception errors result in agents per-

ceiving the opponents' actions with some degree of error (as determined by T and r), though for the implementation error case this perception of error in the opponent's actions is due to the opponent actually making errors in actions. Implementation errors, in addition, result in the agent perceiving its own actions, even when it is a responder, erroneously. In some sense, randomness injected with implementation errors *may subsume* simply having perception error type randomness, though the effect of perception errors is significant for this metric, towards larger bounds. This significance can be seen by comparing Figures 4.2, 4.3 and 4.4 (see Appendix A for further details). Perception errors cause significant deviations from the base line, suggesting randomness injected when considering implementation errors alone adds little to the randomness injected when considering perception errors alone, for larger bounds. On the other hand, there is a significant difference between perception errors alone and implementation errors alone within $0.0001 \leq r \leq 0.001$, and for $n \geq 7$. Moreover, for cases where only implementation errors already show maximum deviation from the base line, perception errors do not add much, suggesting they cannot add more. This suggests implementation errors as they are may be too harsh, and a lower value and finer granularity with them may cover various deviations that perception errors alone covers. A cross between implementation errors and perception errors (using them with different probabilities) may also show more deviations. We leave this investigation for future work.

Considering the average stay within ϵ distance from the equilibrium metric, Figures 4.5, 4.6 and 4.7 suggest that perception errors again cause significant deviations from the base line with increasing bounds. Compared to when agents with both implementation and perception errors are considered, a significant impact is already created by perception errors at $r = 0.00001$, yet implementation and perception errors together lead to a similar impact (against the base line). For $r > 0.00001$, the impact when compared with the base line is harsher with implementation and perception errors together, and with only implementation errors, than with only perception errors. This suggests implementation errors, beyond a threshold do not get affected by perception errors, when both are taken together. This also suggests the implementation errors, below a threshold of boundedness, do not get affected by perception errors, when both are taken together. Imple-

mentation errors alone (Figure 4.5) suggest the same kinds of deviations as when both implementation and perception errors are taken together (Figure 4.7). The randomness injected into the strategies by implementation errors may, in effect, subsume the randomness injected by perception errors, but in a non-linear way, the relationship being something that needs dealing with in future work.

Now taking the case of the average distance from the equilibrium after the first hit as the metric, Figures 4.8 and 4.9 further confirm that implementation errors subsume perception errors, but to what degree is still a question. What we can say is that the distances from the equilibrium achieved by implementation errors (Figure 4.8) and when taking both implementation and perception errors (Figure 4.10) are similar (as with the earlier two metrics), and that the perception errors do not seem to add anything extra to the trends exhibited by simply having agents with implementation errors. On the other hand, perception errors do lead to significant changes in the metric for more complex games and for higher levels of boundedness, as do the other two types (compared to lower levels of boundedness).

When comparing the average payoffs of the players (Figures 4.11, 4.12, 4.13 and 4.14), it is evident, as previously shown, that the deviations with increasing bounds on rationality are smaller for agents with perception errors than they are for agents with implementation errors or for agents with both implementation and perception errors. This again suggests that perception errors do little over having agents with implementation errors already, though perception errors alone do indeed cause changes in the co-evolutionary process and outcomes, and so, do specify a useful notion of bounded rationality, whereas implementation errors alone may do little beyond a certain degree of boundedness (an interesting issue for future research). The effect of perception errors on the maximum and minimum average fitness of the agents are similar across the spectrum of boundedness, however less prominent when compared with agents with implementation errors alone, and agents with both implementation and perception errors.

4.8 Sensitivity of Co-evolution Against Reconciliation Variable Specification

The current specification of our reconciliation variable allows us to explicitly express bounded rationality and understand its role within co-evolutionary simulations, helping us remove the possibility of it being mis-interpreted within simulations, which is the point of the second part of our methodical framework. However, it may still be necessary to understand how sensitive our simulation is, if noise, which partly specifies bounded rationality, and thus the reconciliation variable, were modelled differently. The less sensitive the simulation with respect to variations of the intricacies of the specification, the more generic our specification and conclusions drawn using it may be. Note that our framework does not however rely on studying the sensitivity of simulations in order to help draw general conclusions of how bounded rationality may affect co-evolution. It is still a good exercise, and indeed necessary, if bounded rationality is to be modelled within co-evolution when one may want to match simulation and reality. The framework asks the ACE practitioner to avoid mis-interpretations of socio-economic phenomena of interest within simulations by way of understanding explicitly what the phenomena means within simulations. If indeed co-evolution is sensitive to the specification of the socio-economic phenomena of interest (which is in terms of reconciliation variables), the practitioner must be aware of this.

We consider specifying the *randomness in move (r)* parameter, which partly specifies bounded rationality in our case, differently. Instead of Gaussian noise, whereby we considered noise equivalent to $N(0, r^2)$ injected in the actions, perception, or both, of the agents, depending on the type of their rationality bounds, we now consider the Uniform distribution to generate noise.

Essentially, with probability T, the action or perception will have Uniformly distributed noise within the interval $[-r\sqrt{3}, r\sqrt{3}]$[1] added to it. Mathematically,

[1]We consider r^2 to represent the variance, as for the Gaussian noise model. For a Uniform distribution $U(a, b)$, defined within the interval $[a, b]$, the variance is given by $\frac{1}{12}(b-a)^2$. Since our interval has to lie around zero, because an equal amount of noise may be added to or subtracted from the actions or perception, $a = -b$. This results in $a = -r\sqrt{3}$, and $b = r\sqrt{3}$, which then defines the interval within which uniformly distributed noise with variance r^2, must lie in.

for an agent to be making implementation errors alone would result in the following offers and thresholds:

$$offer = \begin{cases} offer + U(-r\sqrt{3}, r\sqrt{3}) & \text{with probability } T, \text{ and } U(-r\sqrt{3}, r\sqrt{3}) \\ & \text{resampled unless and until} \\ & offer + U(-r\sqrt{3}, r\sqrt{3}) \in [0.0, 1.0], \\ offer & \text{otherwise.} \end{cases} \tag{4.4}$$

$$threshold = \begin{cases} threshold + U(-r\sqrt{3}, r\sqrt{3}) & \text{with probability } T, \text{ and} \\ & U(-r\sqrt{3}, r\sqrt{3}) \text{ resampled} \\ & \text{unless and until} \\ & threshold + U(-r\sqrt{3}, r\sqrt{3}) \\ & \in [0.0, 1.0], \\ threshold & \text{otherwise.} \end{cases} \tag{4.5}$$

An agent to be making only perception errors would not change the offers and thresholds, but instead result in only the offer received from the opponent being erroneous, as follows:

$$received\_offer = \begin{cases} received\_offer + U(-r\sqrt{3}, r\sqrt{3}) & \text{with probability } T, \text{ and} \\ & U(-r\sqrt{3}, r\sqrt{3}) \text{ resampled} \\ & \text{unless and until} \\ & received\_offer + \\ & U(-r\sqrt{3}, r\sqrt{3}) \in [0.0, 1.0], \\ received\_offer & \text{otherwise.} \end{cases} \tag{4.6}$$

For an agent to be making both implementation and perception errors, the

offers, thresholds and the offers received from the opponent will all be erroneous, following Equations 4.4, 4.5 and 4.6.

We observe that the values and the trends in the values of the evaluation metrics across r, T and n are, in general, not significantly different to those observed for the case with Gaussian noise modelling randomness in moves. This is true for all three types of rationality bounds considered in this Chapter. As such, the relationship between the types of rationality bounds, given the evaluation metrics, is the same as for Gaussian noise. Our co-evolutionary simulation is thus not sensitive to specifying randomness in moves using either a Gaussian distribution or a Uniform distribution. For the interested reader, the graphs from the experiments with a Uniform distribution specifying randomness in moves can be found in Appendix B. The experimental setup is exactly the same as for the Gaussian noise case, the only difference being the manner in which r is used, as specified above.

4.9 Future Research Directions

Given our treatment of co-evolutionary simulations in this Chapter, with a tangible handle on them by way of reconciliation variables such that mis-interpretations of socio-economic phenomena be explicitly ruled out from within simulations, there are a multitude of questions and issues that can be raised and directions this research can be taken into. Some of these, and by no means exhaustive, can be listed as follows:

- What does heterogeneity in agent rationality bounds tell us? Heterogeneity can be introduced by considering differences within a type of bound or across types of bounds. For example, one could consider agents with different T or r or both, but the same type of errors in their functioning. One could also consider agents making different types of errors. Heterogeneity is indeed seen as one advantage of using agent based simulations in computational economics, helping model more realistic socio-economic situations, and exploring this idea with our co-evolutionary simulation is one direction for future research.

- Are implementation errors so severe that they become less useful in terms of a change in them changing their effect on the metric in question? Are there diminishing returns on our understanding of how bounded rationality affects co-evolution, from considering implementation errors with greater degrees of randomness?

- Endogenous emergence towards a certain kind of rationality bound(s)? With the way we have quantified bounded rationality, specified by the two parameters T and r, one can easily augment the representation of agents with these parameters, and let evolution take its course, leading to the endogenous emergence of certain kinds or even evolutionary trends on the bounds.

- One can also consider other definitions of bounded rationality, since our methodology does not rely on any one particular definition, e.g. memory use, processing power, historical knowledge, accuracy of foresight [143], knowledge of the other player etc. These may entail the need for a different representation of strategies as compared to the representation considered in this thesis. Furthermore, other representations, e.g. finite state machines where number of states suggests a bound [79], can also be investigated for their effect on co-evolution.

- The actual matching of the solution concept implemented, considering the use of reconciliation variables as suggested here, to specific real world socio-economic situations, for bargaining and other games, is a good candidate for being the immediate next step.

- Can the explicit treatment of bounded rationality using the notion of reconciliation variables lead to simulations that may not have been envisaged but are still interesting? The idea here is for the discovery of simulations that may achieve some desirable computational outcomes, where these outcomes would have been hard to have been thought of as goals towards which the solution concept is to be engineered. The point here would be to facilitate design of co-evolutionary algorithms inspired by socio-economic processes,

to solve computational problems, for example, facilitating the design of algorithms for market based control [30].

4.10 Summary

We propose the notion of reconciliation variables, by matching the phenomenon of interest to an element of co-evolutionary solution concept. This enables a systematic treatment of the phenomenon of interest, which is necessary for ACE research if one is to put a socio-economic interpretation on simulations. We stress that it is premature to interpret simulations in socio-economic terms without challenging the use of and understanding the concerned socio-economic phenomena explaining the co-evolutionary process and outcomes, or else they can get mis-interpreted. A case that we study throughout the thesis is how bounded rationality can be interpreted in a sound manner within simulations. The second part of the framework, which allows for understanding the role of socio-economic phenomena within simulations in explicit algorithmic terms, is what is described and applied in this Chapter. We establish the relationship between the phenomenon of bounded rationality and the co-evolutionary process and outcomes, thus revealing the actual *algorithmic meaning of the phenomenon*, which can guide the explanation of simulations in socio-economic terms. A simple way to look at this is, that if we know what a certain *value* of the phenomenon of interest, given our tangible hold on it, does to the simulation, the task of explaining the workings of the simulations in terms of the phenomenon boils down to explaining the relationship discovered, and prevents avoidable assumption from becoming explanations.

It makes sense to methodically refine a simulation once we have a base line behaviour for it established, as we do by introducing the notion of co-evolutionary solution concepts to ACE research in Chapter 3, unveiling further tangibility by way of reconciliation variables in this Chapter, instead of forcing reality tuned interpretations (thus mis-interpreting socio-economic phenomena within simulations) on the method that does not even achieve the base line behaviour. The *explicit treatment of elements* as compared to intuition, as reality tuned interpretations or indeed mis-interpretations may be called if not challenged for their

validity, will only help advance the field, and makes available a particular kind of scientific lens that ACE research can be looked through. We provide an *explicit framework* for analysing and refining simulations such that socio-economic phenomena of interest explaining the simulations may not get mis-interpreted within these simulations, from the bottom-up point of view of reconciliation variables in this Chapter, in addition to the top-down point of view of co-evolutionary solution concepts in the previous Chapter, and see this as one next step in ACE simulation research, which must not be overlooked.

An important point that we raise in this Chapter is that of engaging in the question of why be mislead by something designed by ourselves. Bounded rationality is a phenomenon that is hard to define or specify in reality, but at least it can be specified and understood in simulation (just like there are specifications in the socio-economic literature), having designed these simulations ourselves. Challenging and understanding the phenomenon in simulation is easier than in reality, so we take the onus of showing how to do it via our methodical framework. We increase the tangibility of co-evolutionary simulations, because it is possible, as shown in this Chapter. The notion behind solutions concepts and reconciliation variables make the simulations tangible from the point of view of bounded rationality. As such, systematic juggling with simulations, with explicit avoidance of mis-interpretations of bounded rationality within them, becomes possible.

Chapter 5

Conclusions

5.1 Summary of Contributions

Looking back at the main point of establishing a link between co-evolutionary algorithm design research and ACE via laying out a methodical framework allowing for the avoidance of mis-interpretations of the algorithms (in socio-economic terms), we now reiterate the contributions that have come about in the process. The first major contribution of this thesis is the framework in itself, that allows for a detailed and holistic look at the socio-economic phenomena being mis-interpreted in simulations. The framework consists of two integral parts that describe it. The descriptions of the integral parts and their application to previous work are contributions of the thesis as well. We state the main idea behind these parts and summarise what we discovered on applying these to previous work, one at a time:

5.1.1 Part 1 of the Framework

The **main idea** behind this part of the framework can be stated as follows:

> *Analysing and refining co-evolution for ACE, using the notion behind co-evolutionary solution concepts, empowered by the existence of top-down solutions, from co-evolutionary algorithm design research. Socio-economic phenomena which may have been mis-interpreted within simulations, can be challenged using this notion, and their mis-interpretation revealed. As such, analysis and refinements of co-evolutionary*

simulations, guided by the presence of top-down solutions, allow for a ***top-down avoidance of mis-interpretations*** *of socio-economic phenomena of interest within simulations.*

The **application** of the above to previous work is now summarised:

The implemented co-evolutionary solution concept from previous work [58] was looked at. This co-evolutionary solution concept took bounded rationality, a socio-economic phenomenon, for granted, assuming it to be causing deviations from the expected behaviour of the algorithm, without scrutiny. We showed that by working on the elements of the implemented co-evolutionary solution concept systematically, and scrutinising as to what may have caused the deviations from the expected behaviour algorithmically, we could say the following:

- The deviations were being caused by the improper implementation of the co-evolutionary solution concept. We considered the elements of the co-evolutionary solution concept one at a time. We found that the variation operator was being implemented incorrectly. Selection of infeasible opponents was being carried out, which became clear by looking at the interaction details of the agents. Moreover, local evaluation of the strategies, lead to disruptions in the strategies due to the localisation of selection pressure. These are algorithmic details which were being overlooked and termed as bounded rationality.

- Changing the elements of the solution concept, more specifically, correcting the variation operator, correcting for infeasible opponent selection at the interaction level and establishing three rules that govern the evaluation of strategies, we were able to address the issues mentioned above. We were also able to show that monotonicity in the concessions made by agents was not enough for the algorithm converging to the equilibrium. A closer and systematic look at the co-evolutionary solution concept thus lead to the algorithm converging to the subgame perfect equilibrium solution, while the agents remained boundedly rational. We thus obtained a new implemented co-evolutionary solution concept that did indeed achieve the envisaged goals, with boundedly rational agents.

In essence, we understood in explicit algorithmic terms, what the assumption of bounded rationality actually meant in previous work. Moreover, we showed that for bargaining games with one issue and $n \geq 1$, the algorithm would, practically speaking, indeed converge to the equilibrium solution, though the function evaluations needed would indeed increase with increasing n. In what way these evaluations increase is something we will consider understanding in the future. More importantly, this investigation of the algorithm through the lenses of co-evolutionary solution concepts allows us to say that previous work was not implementing the envisaged solution concept. The importance of the notion behind co-evolutionary solution concepts for ACE research thus became clear.

In addition, we discovered what bounded rationality did not mean in the context of socio-economic simulations using co-evolutionary algorithms. We stress that, although we discover the issues and propose remedies within the realm of bargaining games and the co-evolutionary algorithms considered in this thesis, the method of understanding the simulations through the lenses of co-evolutionary solution concepts (our first part of the framework) is *generally applicable*. We show that *challenging unassessed assumptions* about socio-economic phenomenon, carried in to co-evolutionary simulations, using the notion behind co-evolutionary solution concepts, is indeed a methodology worth considering for ACE research. The general practice within ACE of implicitly assuming bounded rationality within simulations, thus leaving it unscrutinised, is also called into question. We believe that these revelation are significant enough for ACE practitioners to *re-think current practice* and pay heed to our framework. In fact, going one step further, and leaving no room for mis-interpretations of socio-economic phenomena, in our case bounded rationality, by explicitly considering its relationship with the co-evolutionary simulations, leads to the next part of our methodical framework.

5.1.2 Part 2 of the Framework

The **main idea** behind this part of the framework can be stated as follows:

> *Analysing and refining co-evolution for ACE, using the notion behind reconciliation variables proposed in the thesis. Reasonably associating mis-interpreted socio-economic phenomena of interest with the ele-*

ments of the implemented co-evolutionary solution concept, parametrising and quantifying the elements, we obtain our reconciliation variables. Systematically analysing the simulation for its relationship with the reconciliation variables or for its closeness to desired behaviour, using this parametrisation, is the suggested idea. Analysis and refinements based on such an explicit expression of the socio-economic phenomena of interest, allow for a ***bottom-up avoidance of mis-interpretations*** *of the phenomena within simulations.*

The **application** of the above, to the co-evolutionary simulation refined using the first part of the framework, is now summarised:

We wanted to understand what bounded rationality really meant to socio-economic simulations, since we discovered what it did not mean (algorithmically) and that the agents were still boundedly rational. It turned out that phenomenon of interest like bounded rationality do not have an obvious algorithmic representation within co-evolutionary simulations. This is in contrast to phenomena like *information exchange* or *innovation* for instance, wherein, variation operators, viz. crossover and mutation respectively, have been used to interpret them in the past [1]. However, if we are indeed interested in simulating socio-economic learning where phenomena of interest like bounded rationality do play a role in the real world, specifically if we want to avoid their mis-interpretation within simulations, then we need to have a way to tangibly understand them, rather than saying that co-evolutionary algorithms simulate bounded rationality by definition (which is general practice). From the point of view of ACE research, if a simulation has to have predictive power, it needs to be designed in a methodical manner, as opposed to having mis-understood characteristics (mis-interpreted socio-economic phenomena) from the outset.

Our goal was to see if the implemented co-evolutionary solution concept can indeed be related to socio-economic phenomena of interest such that the phenomena may not get mis-interpreted and a systematic study of the relationship that the phenomena have with the co-evolutionary algorithm, be carried out. *In other words, we wanted to make co-evolutionary algorithms tangible from the point of view of the socio-economic phenomena under question.* The notion behind co-evolutionary solution concepts allowed us to view the algorithm as elements which

could be tangibly handled. The question then was, as to how to associate the phenomena of interest with the elements.

We showed how bounded rationality can be associated with the elements of the implemented co-evolutionary solution concept. This was done by finding *reasonable* explanations of the phenomenon in socio-economic literature, and the elements that were most likely to implement those explanations. We thus had a link. We associated bounded rationality with the representation, given that socio-economics literature suggests what is known as a *trembling hand*, with which agents make moves. The trembling hand suggests that, with some probability, the agents will make errors in the moves they make. This could directly be associated with the representation of strategies. We were thus able to define three types of bounded rationality: *implementation errors*, *perception errors* and a *combination of the two*. We modelled these using two parameters, viz. *trembling probability (T)* and *randomness in moves (r)*, within the representation. As such, we got a firm hold on the malleability of bounded rationality in explicit quantitative and algorithmic terms. Bounded rationality thus becomes, what we call, our *reconciliation variable*.

We then defined the characteristics of the implemented co-evolutionary solution concept that we were interested in understanding, given the range within which bounded rationality could now be systematically varied. These included characteristics like: the *average first hitting time* of reaching within ϵ distance from the equilibrium, the *average stay* within ϵ distance from the equilibrium, the *average distance from the equilibrium* over time after the first hit, the *average payoff*, the *average maximum and average minimum payoffs*.

A *methodology* was thus presented, which allowed for a *tangible and systematic handling* of the co-evolutionary algorithm from the point of view of bounded rationality (our phenomenon of interest), i.e. tangibly varying bounded rationality (varying T and r, given one of the three types of bound), and analysing the relationship that the algorithm had with it was thus made possible. With this methodology, we could then say:

- One view of bounded rationality is not necessary to be used with the methodology, as long as one can reasonably associate bounded rationality with the elements of the co-evolutionary solution concept. One could

work out the associations a particular view of bounded rationality in economics has with the elements of the co-evolutionary solution concept, and understand how this view affects the process and outcomes. As such, the methodology does not rely on one view or explanation of the socio-economic phenomena of interest.

- Multiple metrics or characteristics of the algorithm are necessary to be scrutinised if one is interested in understanding how a phenomenon of interest affects the simulations. For example, only considering payoffs (and not considering another metric related to convergence characteristics), is not enough to conclude about how the phenomenon of interest affects the simulations, and may lead to misleading conclusions about the relationship.

- One can compare different views of bounded rationality and carry out further investigation as to which view may be more relevant to the kind of simulation wanted. For instance, we show that perception errors may be subsumed by implementation errors. Perception errors do not seem to add anything extra to the trend exhibited by simply having implementation errors, though they form a useful notion of bounded rationality, as they do change the base line behaviour of the algorithm significantly. Implementation errors may do little beyond a certain degree of boundedness and is a line of work worth considering in the future.

- Even a difference like the amount of noise injected into the strategies (as done with implementation errors, perception errors and a combination of the two) results in differences in the outcomes of the co-evolutionary algorithm, suggesting the *importance* of studying the phenomena of interest in a methodical and explicit manner (and not assuming it within simulations) for the purposes of ACE research.

- As the problem or game becomes more complex, a change in bounds changes the behaviour of the algorithm. A systematic study can provide with further information as to what regions of bounds may be useful for the ACE application at hand (for example, which regions may not change the behaviour of the algorithm with increasingly complex games), specially if the idea is

to ensure predictive content from the algorithm as the problem becomes more complex.

The explicit consideration of bounded rationality within the simulation and understanding how it affects the simulation, in itself, is an exercise missing from ACE literature and thus a contribution. It is worth considering because we can now be *clear about the algorithmic meaning of the phenomenon of interest and the way it affects co-evolutionary simulations* instead of making unassessed assumption about the phenomenon. The fact that makes the explicit consideration of the phenomenon a reasonable thing to do, is our proposal of reconciliation variables, which suggests for the phenomenon to be *reasonably associated* with the elements of the co-evolutionary solution concept that specifies the simulation, i.e. must have a backing in socio-economic literature (in our case it is literature on bounded rationality) and then associated with the elements. Given this backing, one can at least be sure of the inner workings of the phenomenon, and this can follow a systematic study.

5.2 Further Observations Regarding Our Framework

Our methodology can also be seen as *complementing a lot of the significant research efforts* in the direction of disciplining the use of co-evolutionary algorithms within ACE. Taking the significant efforts in disciplining co-evolutionary algorithms within ACE one at a time, we now show how our methodology can inform the ACE community in the following complementary ways:

- The fundamental methodology for designing co-evolutionary simulations needs attention, as mentioned in [2], and is one alternative to understanding and improving the quality of simulations for socio-economic problems. In [2], evolutionary algorithm parameters that do not have a direct socio-economic interpretation were investigated in order to understand the robustness of two co-evolutionary solution concepts, when these parameter values are varied. However, we focus on a methodology that helps parametrising

socio-economic phenomena of interest by associating them with the algorithm, and parametrising the elements that make up the algorithm, i.e. we parametrise the algorithm in terms of the socio-economic phenomena. This enables understanding the relationship between the socio-economic phenomenon of interest and the simulation, which may further guide moulding the algorithms for socio-economic simulations in a methodical manner, if needed. In effect, [2] helps removes the influence of algorithmic parameters, which cannot be understood in socio-economic terms, on simulations, and we introduce algorithmic parameters, which allow socio-economic terms to be understood, in simulations.

- Unaware of the existence of the notion behind co-evolutionary solution concepts, a study was undertaken in [8]. There, the main issue dealt with was to understand the changes in the genetic algorithm learning model [5] needed, so that it can be refined towards selecting a particular type of equilibrium, which was otherwise not possible. Systematically changing the genetic algorithm setup, guided by another co-evolutionary solution concept i.e. classifier systems, resulted in the genetic algorithm converging to the equilibrium. This work essentially echoes the need for viewing co-evolutionary algorithms through the lenses of co-evolutionary solution concepts. Whereas they modified the co-evolutionary solution concept, working on the fitness evaluation systematically, guided by the conflicting results obtained using another solution concept, we simply look at the elements of the co-evolutionary solution concept systematically, and obtain equilibrium selection. Co-evolutionary solution concepts, if explicitly examined, are thus a powerful way of refining the algorithm.

- The need for a mathematical theory which may explain the results of simulations using genetic or more generally, evolutionary algorithms, is stressed in [39]. Without it, predicting the behaviour of evolutionary algorithms without actual simulations becomes impossible. Using Markov chains [105], a study was indeed carried out in [39] along these lines. Whereas this is interesting from the point of view of understanding what a chosen evolutionary algorithm may result in when used as a simulation, we lay out a

framework for making a methodical choice of the evolutionary algorithm. Whereas an off-the-shelf evolutionary algorithm is treated theoretically in [39], we envisage our methodology being used to help narrow down on to an algorithm first, by way of analysing and refining the algorithm as suggested by our framework, which may be followed by a theoretical analysis for further confidence.

The main weakness in our methodology is that it is entirely empirical, and so, may lead to further confidence considerations in the resultant refined co-evolutionary solution concept. Although the methodology shows how to understand the effects of socio-economic phenomena on co-evolution in a methodical manner, a proper validation against real experimental data for socio-economic games is required so that actual simulations of real socio-economic situations be derived. If the scrutinised phenomenon of interest warrants the simulation to match reality, which is a necessity for real world application, the validation must indeed be carried out. From the point of view of our methodology, this should be done once we have an algorithm designed towards achieving a base line behaviour, and we have the reconciliation variables identified to allow for moulding the co-evolutionary solution concept being implemented by the algorithm.

In this thesis, in essence, we lay out an *explicit holistic framework* for studying socio-economic phenomena of interest such that they may not get mis-interpreted within simulations, and thus analysing and refining simulations from the point of view of co-evolutionary solution concepts and reconciliation variables. In effect, we show how socio-economic phenomena of interest can be scrutinised and indeed understood within co-evolutionary algorithms from the top down and from the bottom up respectively, in a methodical manner. For the top-down avoidance of mis-interpretations of the phenomena, we showed how the notion behind co-evolutionary solution concepts can be used by the ACE research community to challenge the phenomena within simulations and simulate socio-economic outcomes of interest (convergence to the equilibrium). For the bottom-up avoidance of mis-interpretations of the phenomena, proposing the notion behind reconciliation variables, we showed how the phenomena of interest can be explicitly expressed within these algorithms such that a systematic study of such phenomena be carried out in algorithmic terms (understanding the effect of bounded ratio-

nality on co-evolution). From the ACE simulation research perspective, this is one next step which must not be overlooked.

5.3 Future Work

Using the methodology described in this thesis, given the problem of simulating learning and adaptation in bargaining games, there are lots of research questions and directions that can further be explored, some of which are:

- What does heterogeneity in agent rationality bounds tell us? We can further explore bounded rationality along the lines of agents being differently bounded. Real life bargaining falls into this category and heterogeneity is one of the strong motivations behind the usefulness of ACE.
- What can the endogenous emergence towards a certain kind of rationality bound(s) tell us? Again, further exploration of the co-evolutionary solution concept can be carried out in order to understand if there are stable rationality bounds that may emerge, or if not, then to understand how the bounds evolve over time.
- What can other definitions of bounded rationality tell us? We can use other explanations of bounded rationality in the literature and understand how those explanations affect the co-evolutionary solution concept to further our understanding of bounded rationality in socio-economic simulations.
- Can we discover what-if simulations? It may be that we may discover what-if simulations that allow for their usage in the computational realm (instead of being models of socio-economic learning) e.g. computational resource allocation that warrant certain desirable outcomes, which may be hard to engineer otherwise.
- How well does the implemented co-evolutionary solution concept generalise to more complex games? Scalability of socio-economic simulations to games where it is cumbersome or even intractable to come up with an analytical solution is an important motivation behind ACE, and researching into the

scalability of the implemented co-evolutionary solution concepts to more complex games is indeed an important research direction.

Appendix A

Empirical Results From Chapter 4: Randomness in Moves (r) Specified Using a Gaussian Distribution

A.1 Overview

In Chapter 4, we lay out a methodology for a proper consideration of co-evolutionary solution concepts within ACE, by proposing what we call, reconciliation variables. There are socio-economic phenomena which may not have an obvious relationship with the co-evolutionary algorithm or indeed, the co-evolutionary algorithm may not have an obvious socio-economic (in terms of the phenomena) interpretation. Generally though, for socio-economic phenomena, even if they have an obvious or intuitive link with the algorithm, one should approach (scrutinise and understand) this link in a systematic way, simply for the simulation to be a good approximation to the socio-economic situation being simulated.

We show the link that bounded rationality has with the co-evolutionary algorithm in Chapter 4. We do this by associating it with an element (representation) of the co-evolutionary solution concept, parametrising and quantifying the element thus associated. This follows the understanding of the relationship between the, now algorithmically tangible phenomenon of bounded rationality (our reconciliation variable), and the characteristics (in the form of certain evaluation

metrics) of the co-evolutionary algorithm one may be interested in investigating with respect to bounded rationality. Here, we show the empirical results in detail that were obtained when establishing this relationship, given that the graphs in Chapter 4 were summary graphs.

We consider the types of bounded rationality, showing the results for each evaluation metric, given the game (specified by a value of $n \in [1, 10]$, where n is an integer), and the bound on the rationality. Section A.2 shows the *averages* across 50 runs, together with the standard error at a 95% confidence level. Note that in the graphs, both the averages and standard error values are shown as *colour*, where the colour of the '+' sign shows the standard error (95% confidence). The colour can be referred to using the colour map on the left of each graph to understand the values for these two. Section A.3 shows the magnitude of the standard errors (95% confidence) again, but separately for each metric, game and bound on rationality, given the type of bounded rationality. The '+' indicators in the graphs in Section A.2 are essentially shown to get an idea of the error *on the same scale* as the metric.

A.2 Averages with Confidence

We show the values for various metrics considered in Chapter 4, which were all averages, for the three types of bounded rationality investigated. For each type and for each evaluation metric, we show these values for the evaluation metric on a graph, that caters for all the rationality bounds and games investigated in that type. Note that each type of rationality was quantified by two parameters T and r, which are the *trembling hand probability* and the *randomness in move*. These two, together with n are the axes in the graphs.

Assuming that the values for each metric, from the $N = 50$ runs/samples of the algorithm, for each *game and rationality bound setting* came from a *t-distribution*, we can calculate the standard error and the confidence intervals for the average to lie within 95% of the area under the distribution, around the sample average.

The standard error is given by $s_m = s/\sqrt{N}$, where s is the sample standard deviation, which in turn is given by $s = \sqrt{\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{N-1}}$, μ being the sample average.

A t value of 2.0096 gives the amount of standard error on either side of the mean μ that could be expected, for the average to lie within 95% of the area under the distribution with sample average μ. This is given by $\mu \pm 2.0096 s_m$. We show the amount of standard error i.e. $2.0096 s_m$ for each game and rationality bound setting in the graphs as well, indicated by the colour of the '+' sign. The errors are low. They are similar when seen at the scale of the metric, and thus appear to be of the same colour. To get a better idea of these, Section A.3 plots the *magnitude* of each on graphs similar to the ones considered in this Section. The graphs are presented in the same order as in this Section for ease in referring them.

To guide the eye, and for easy of use of these graphs, we suggest looking at the '+' indicators on the squares (squares representing the values of the metric) on the graphs in this Section by determining their value from the colour bar beside each graph (graphs in Section A.3 will give a more precise value for the interested reader). Adding and subtracting the value to the value of the square (determined using the colour bar again) gives the confidence interval. Comparing two such squares for significance amounts to seeing whether or not their confidence intervals overlap. If they do not overlap, the difference is significant with a confidence of 95%.

A.2.1 Implementation Errors

Average first hitting time

Figure A.1 shows the *average first hitting time*, across 50 runs for all values of r, T and n, for implementation errors.

Average stay in equilibrium

Figure A.2 shows the *average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n, for implementation errors.

Average distance from equilibrium

Figure A.3 shows the *average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for implementation errors.

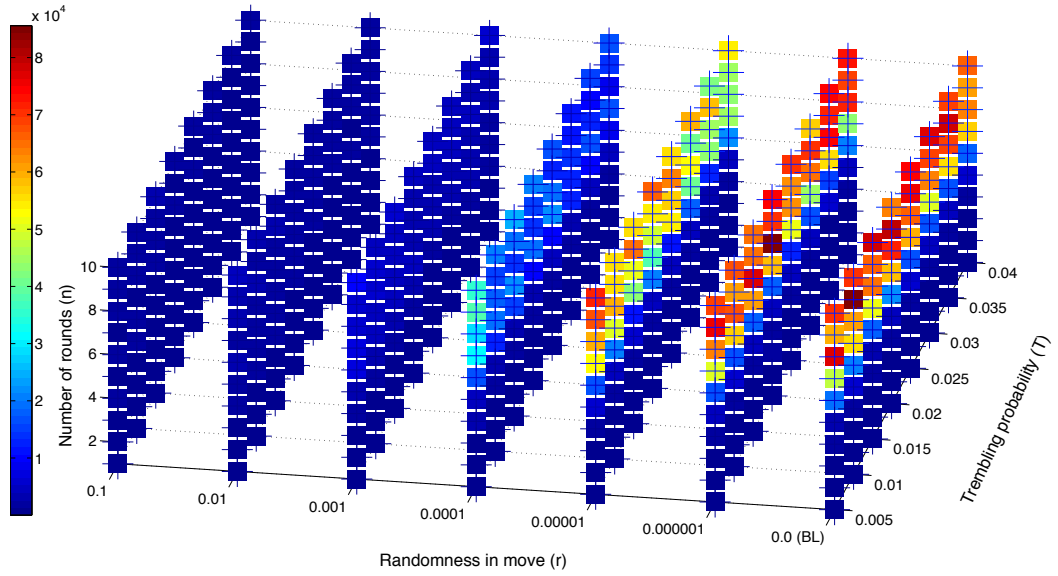


Figure A.1: *Average first hitting time*, across 50 runs for all values of r, T and n (implementation errors). See Figure A.46 for the magnitudes of errors in detail.

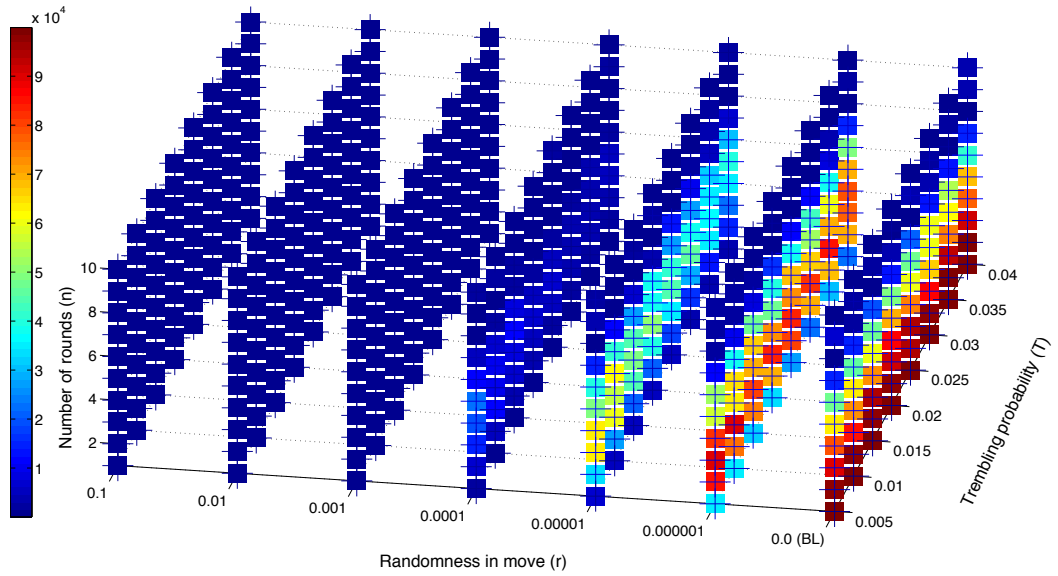


Figure A.2: *Average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n (implementation errors). See Figure A.47 for the magnitudes of errors in detail.

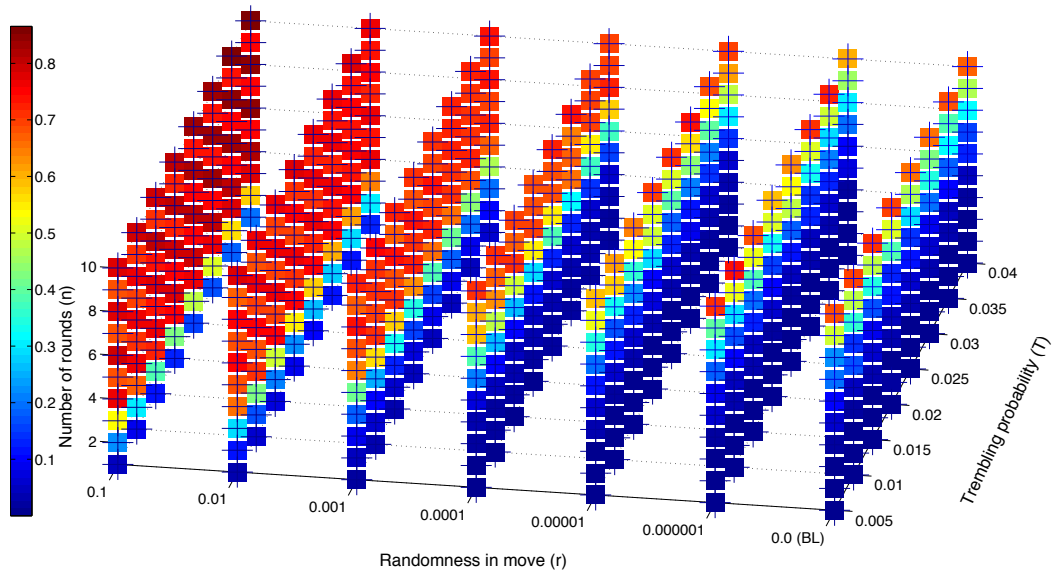


Figure A.3: *Average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (implementation errors). See Figure A.48 for the magnitudes of errors in detail.

Average payoff

Figures A.4 and A.5 show the *average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Figures A.6 and A.7 show the *average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Average maximum payoff

Figures A.8 and A.9 show the *average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Figures A.10 and A.11 show the *average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

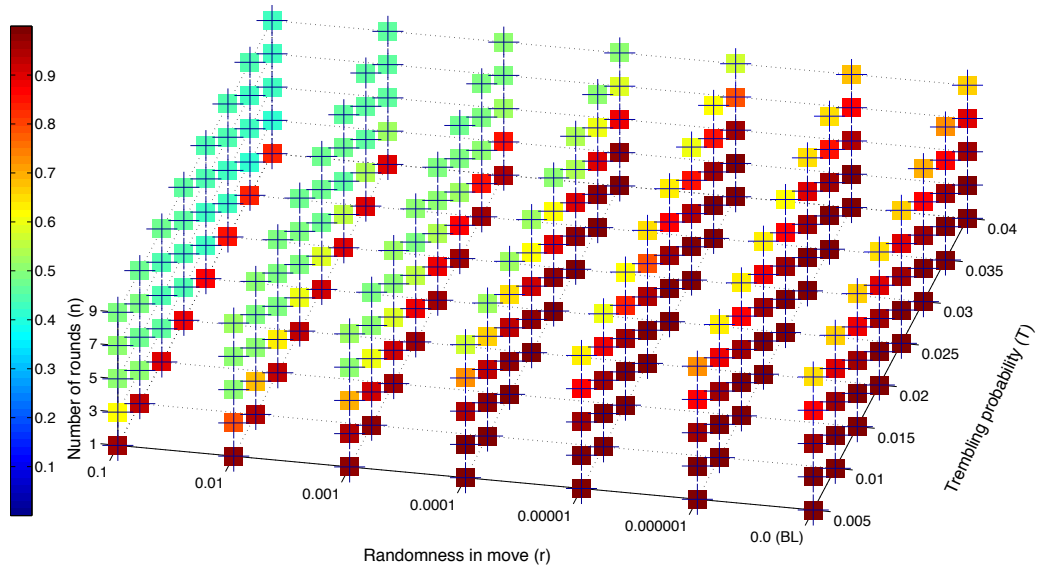


Figure A.4: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure A.49 for the magnitudes of errors in detail.

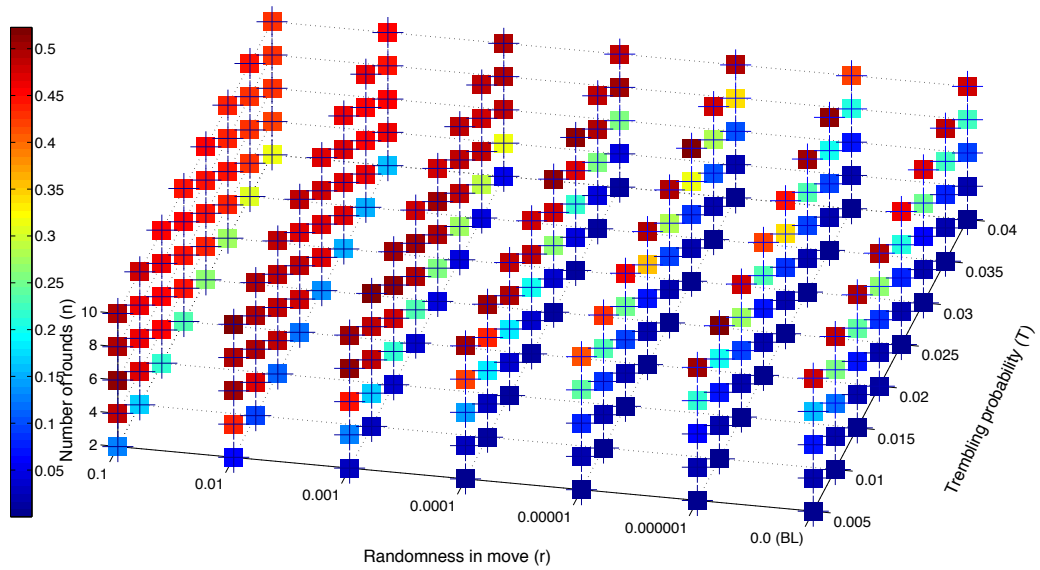


Figure A.5: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure A.50 for the magnitudes of errors in detail.

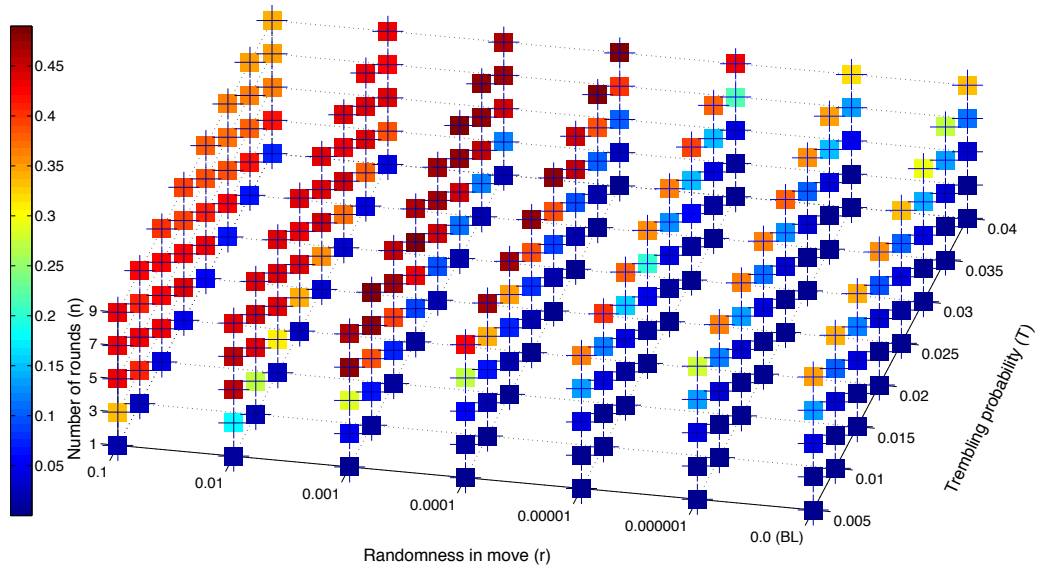


Figure A.6: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure A.51 for the magnitudes of errors in detail.

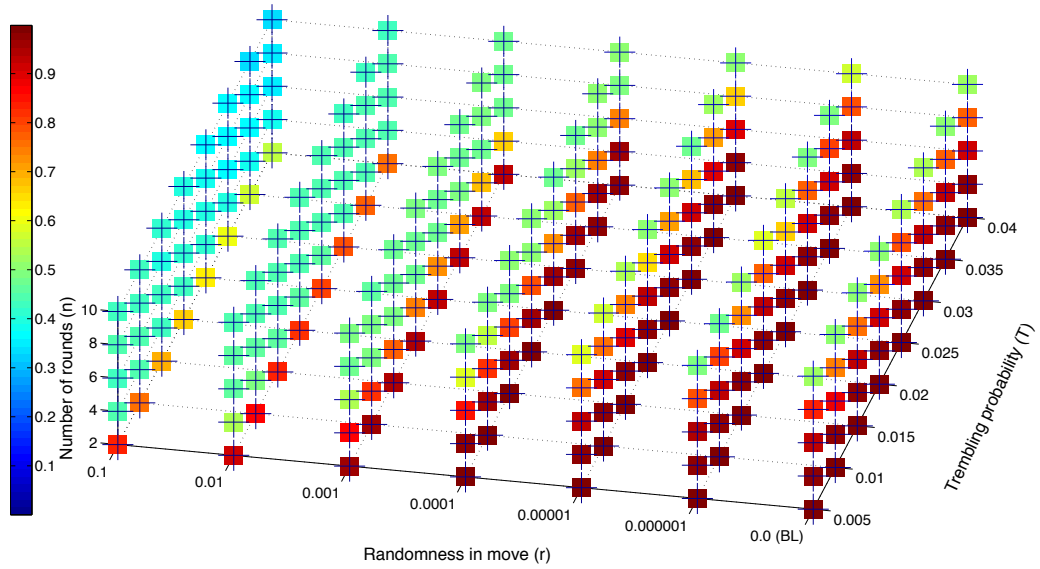


Figure A.7: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure A.52 for the magnitudes of errors in detail.

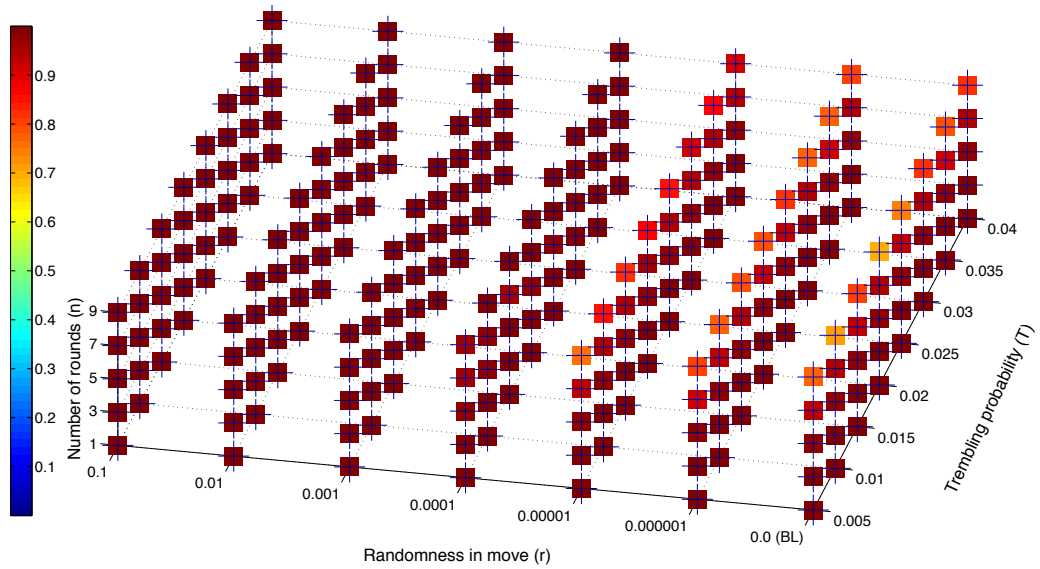


Figure A.8: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure A.53 for the magnitudes of errors in detail.

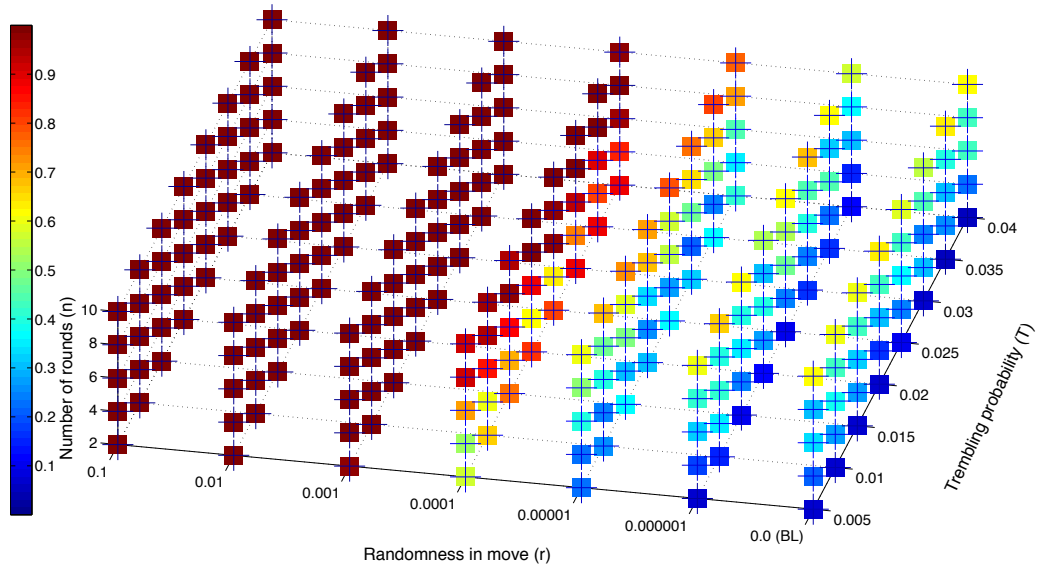


Figure A.9: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure A.54 for the magnitudes of errors in detail.

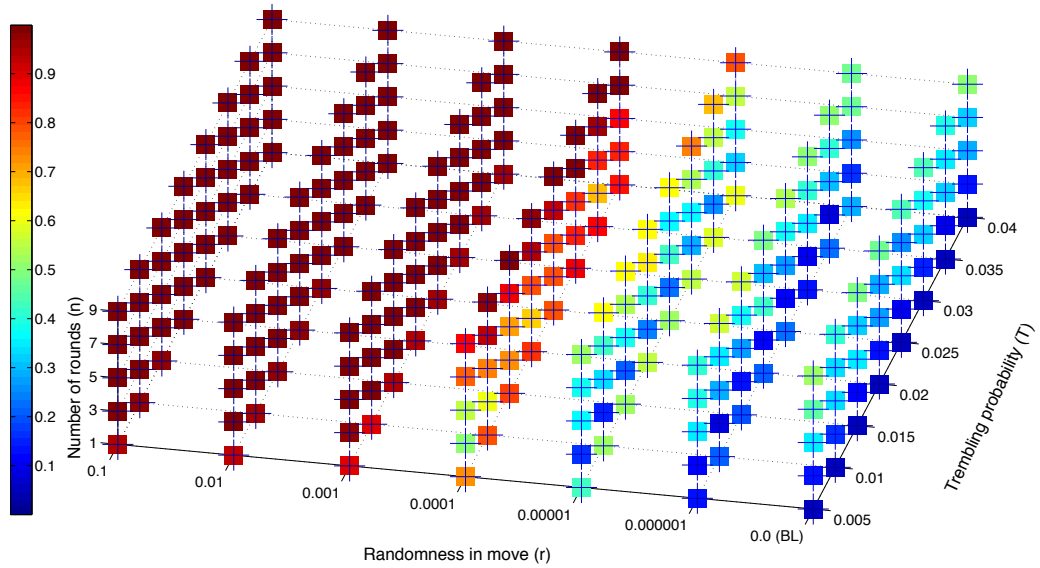


Figure A.10: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure A.55 for the magnitudes of errors in detail.

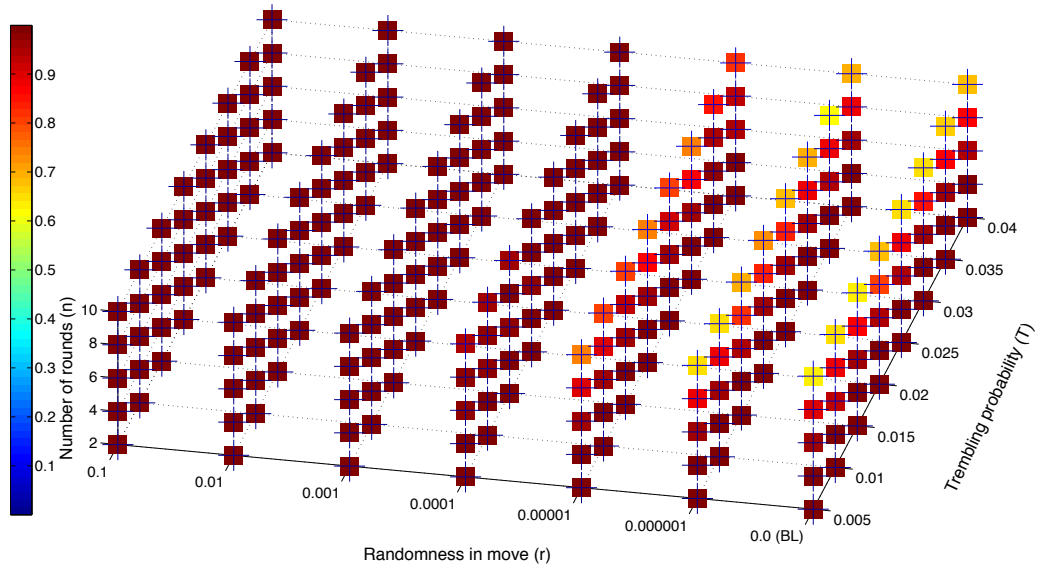


Figure A.11: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure A.56 for the magnitudes of errors in detail.

Average minimum payoff

Figures A.12 and A.13 show the *average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

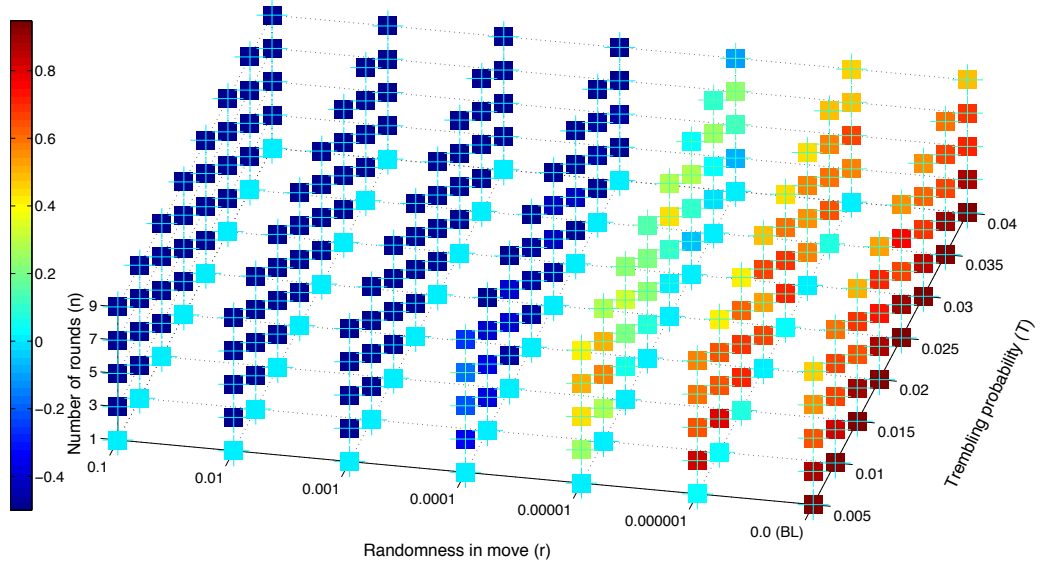


Figure A.12: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure A.57 for the magnitudes of errors in detail.

Figures A.14 and A.15 show the *average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

A.2.2 Perception Errors

Average first hitting time

Figure A.16 shows the *average first hitting time*, across 50 runs for all values of r, T and n, for perception errors.

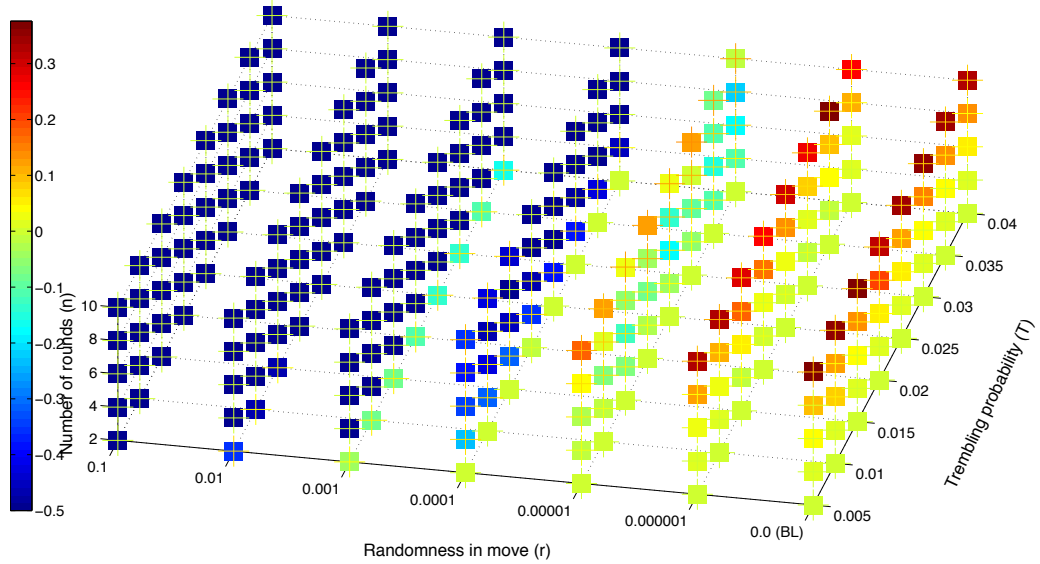


Figure A.13: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure A.58 for the magnitudes of errors in detail.

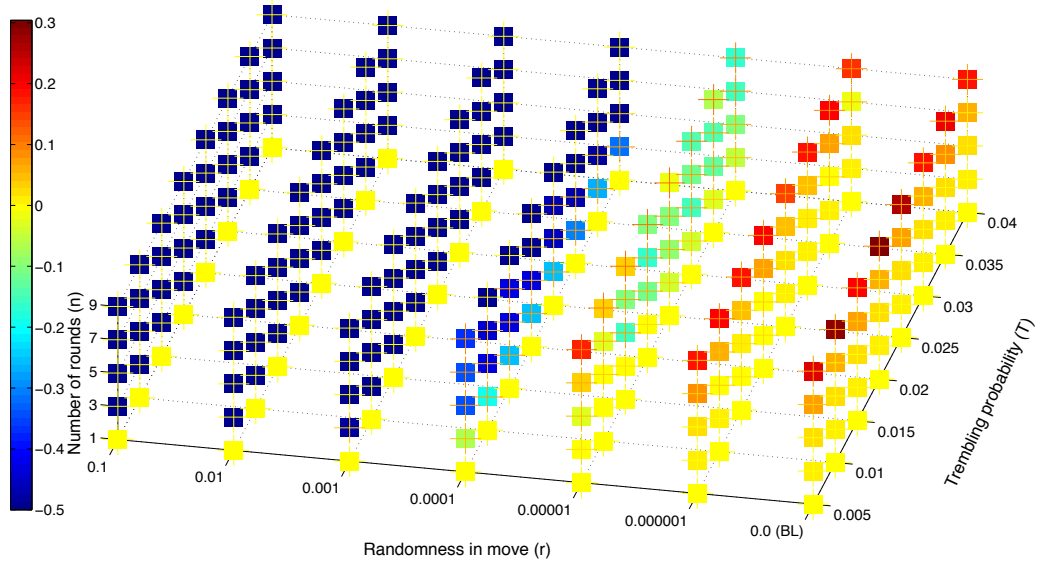


Figure A.14: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure A.59 for the magnitudes of errors in detail.

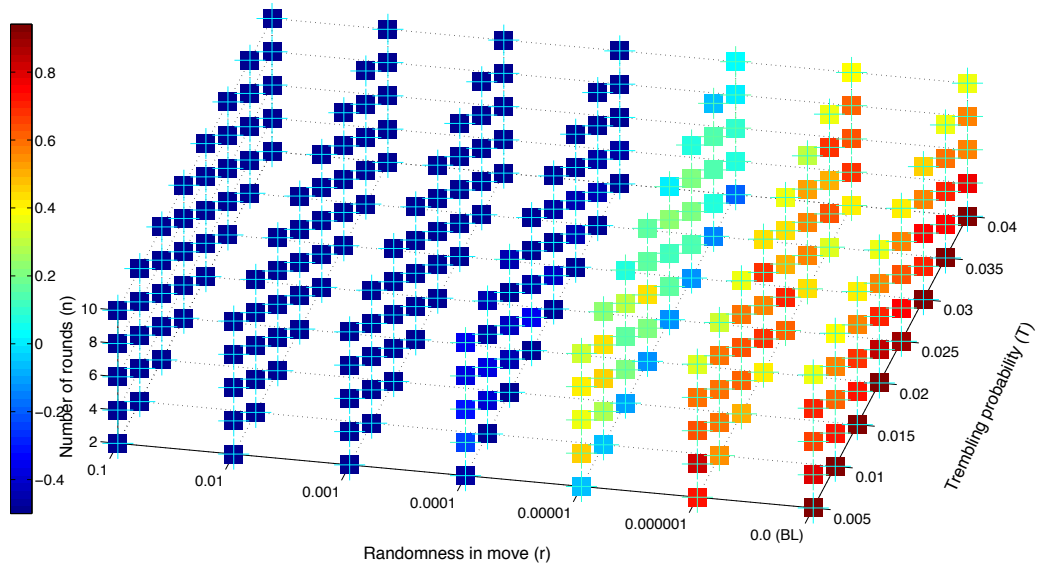


Figure A.15: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure A.60 for the magnitudes of errors in detail.

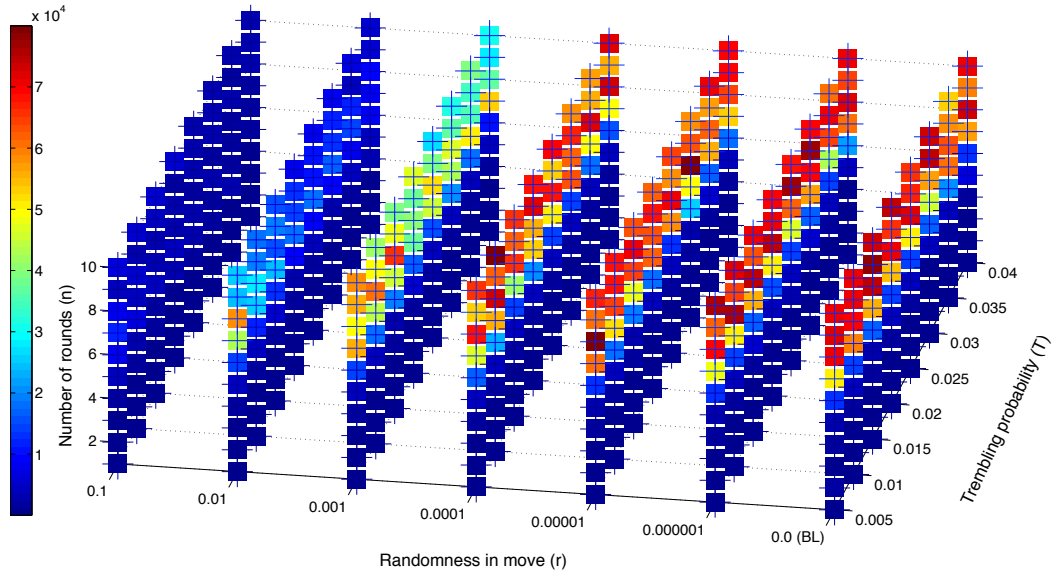


Figure A.16: *Average first hitting time*, across 50 runs for all values of r, T and n (perception errors). See Figure A.61 for the magnitudes of errors in detail.

Average stay in equilibrium

Figure A.17 shows the *average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n, for perception errors.

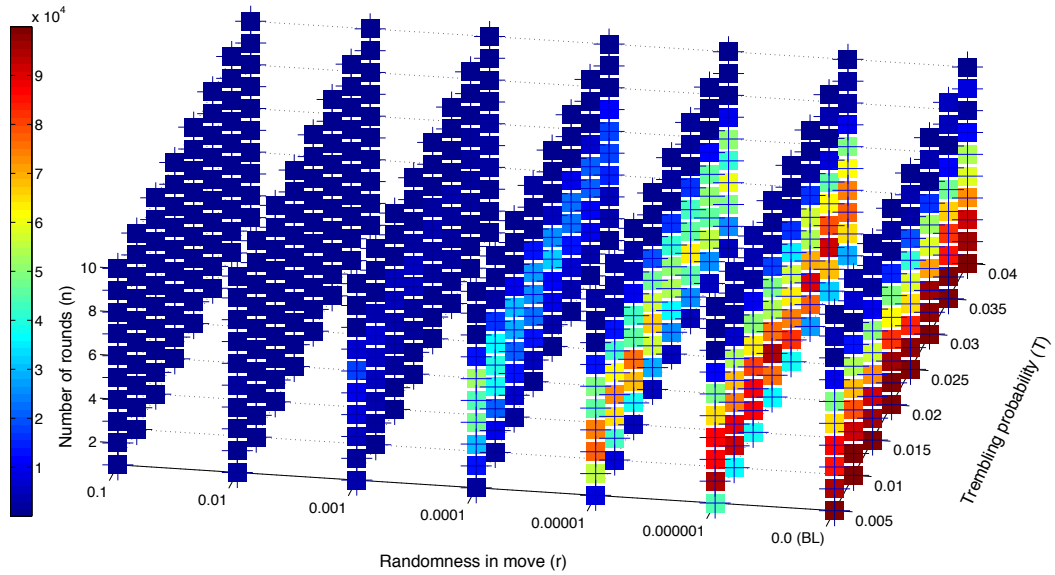


Figure A.17: *Average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n (perception errors). See Figure A.62 for the magnitudes of errors in detail.

Average distance from equilibrium

Figure A.18 shows the *average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for perception errors.

Average payoff

Figures A.19 and A.20 show the *average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures A.21 and A.22 show the *average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

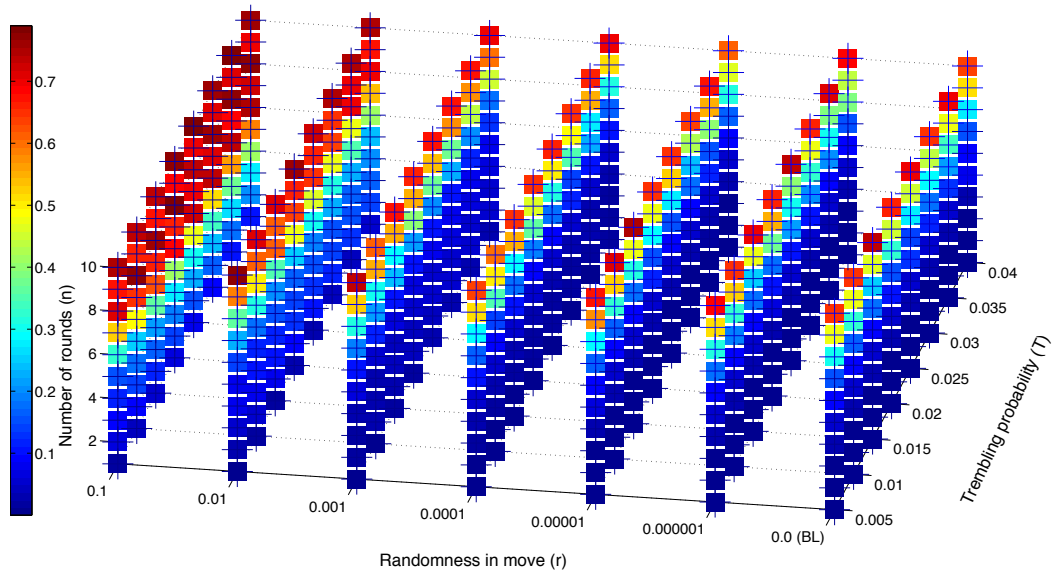


Figure A.18: *Average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (perception errors). See Figure A.63 for the magnitudes of errors in detail.

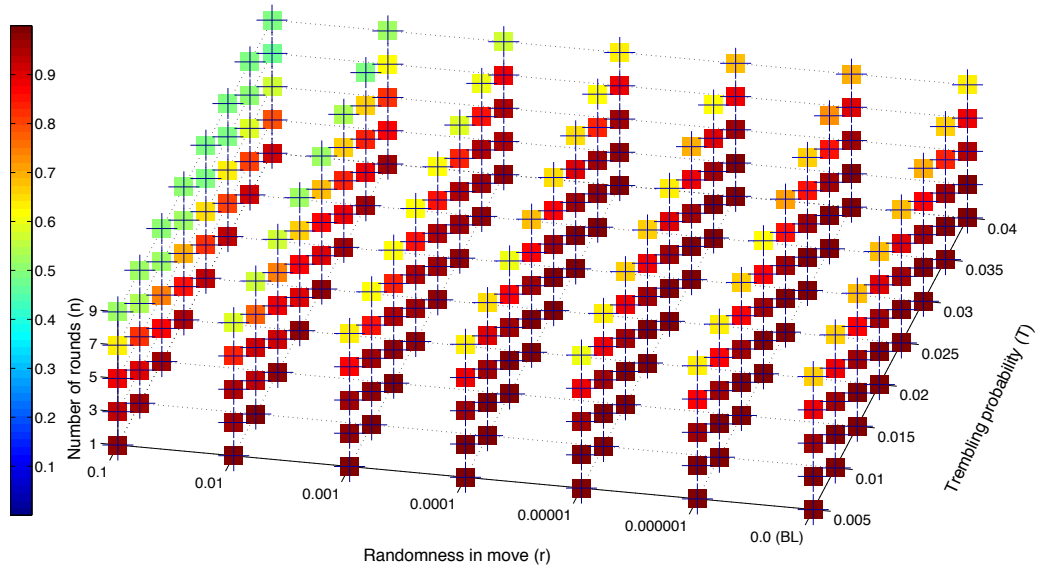


Figure A.19: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure A.64 for the magnitudes of errors in detail.

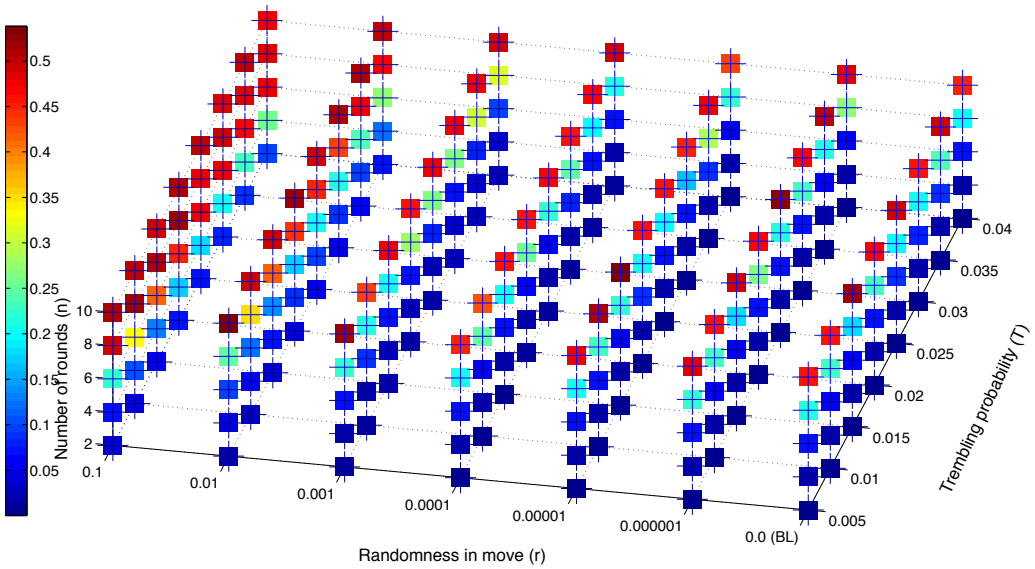


Figure A.20: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure A.65 for the magnitudes of errors in detail.

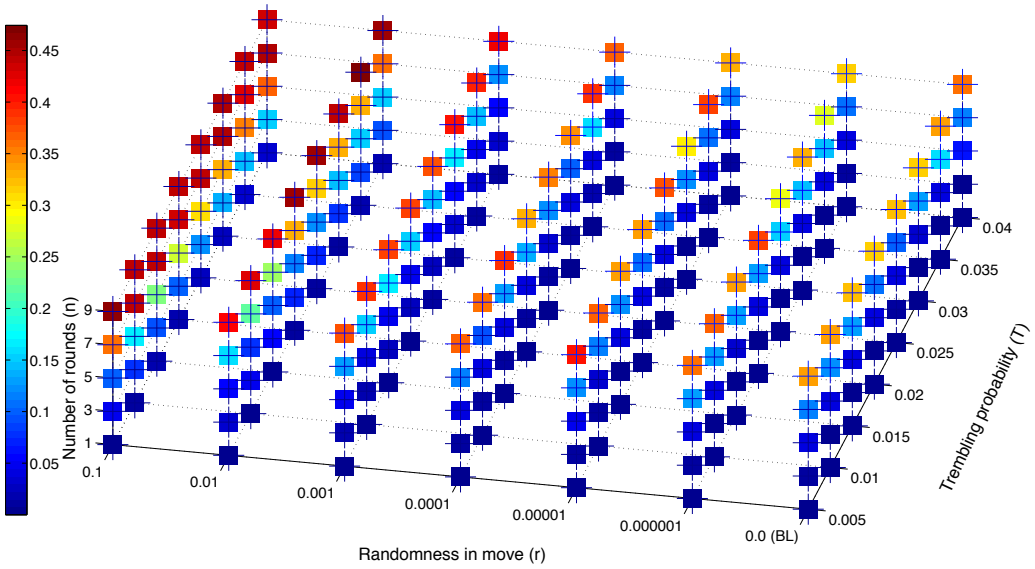


Figure A.21: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure A.66 for the magnitudes of errors in detail.

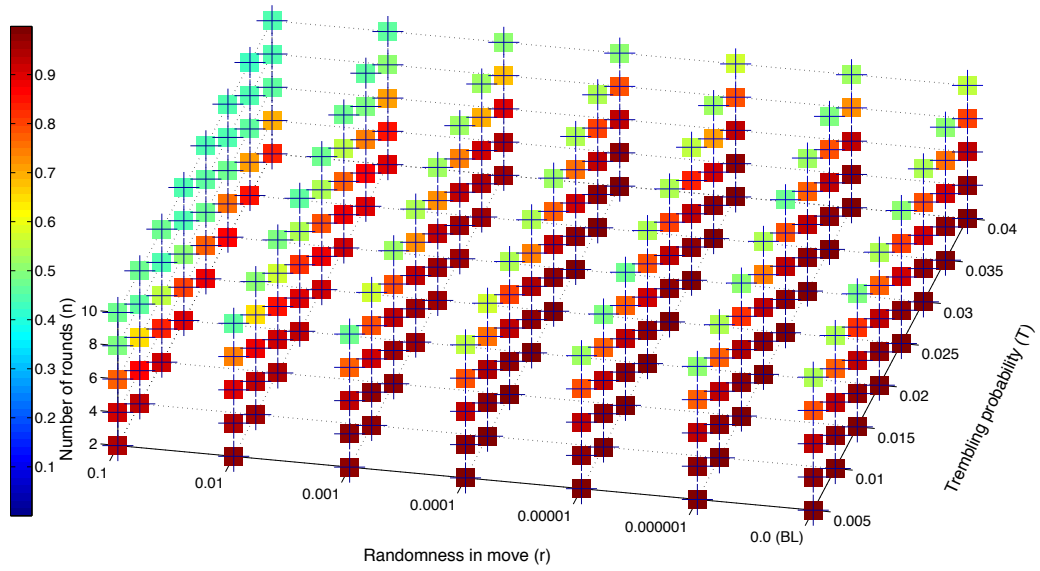


Figure A.22: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure A.67 for the magnitudes of errors in detail.

Average maximum payoff

Figures A.23 and A.24 show the *average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures A.25 and A.26 show the *average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Average minimum payoff

Figures A.27 and A.28 show the *average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures A.29 and A.30 show the *average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

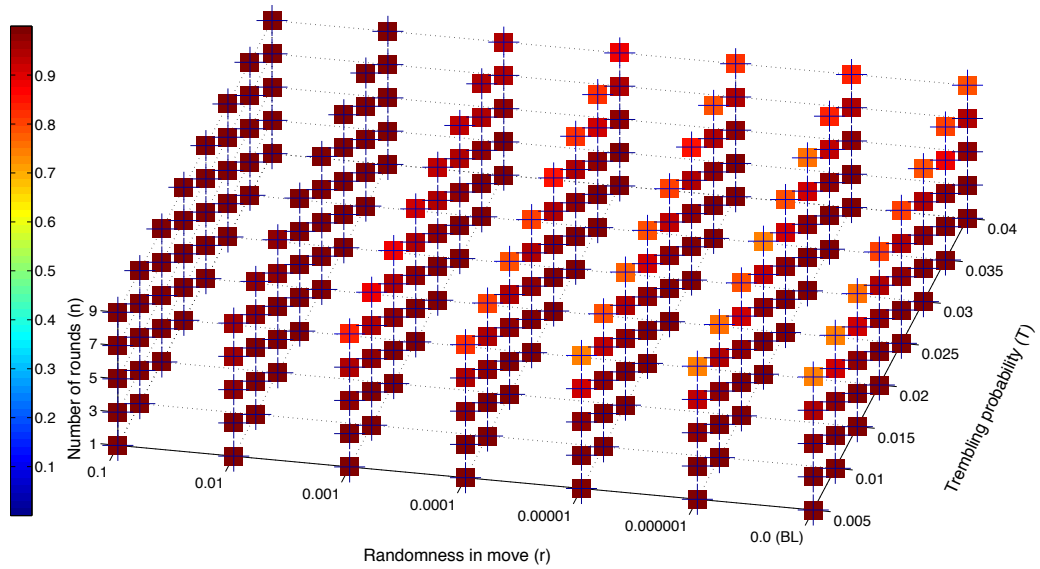


Figure A.23: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure A.68 for the magnitudes of errors in detail.

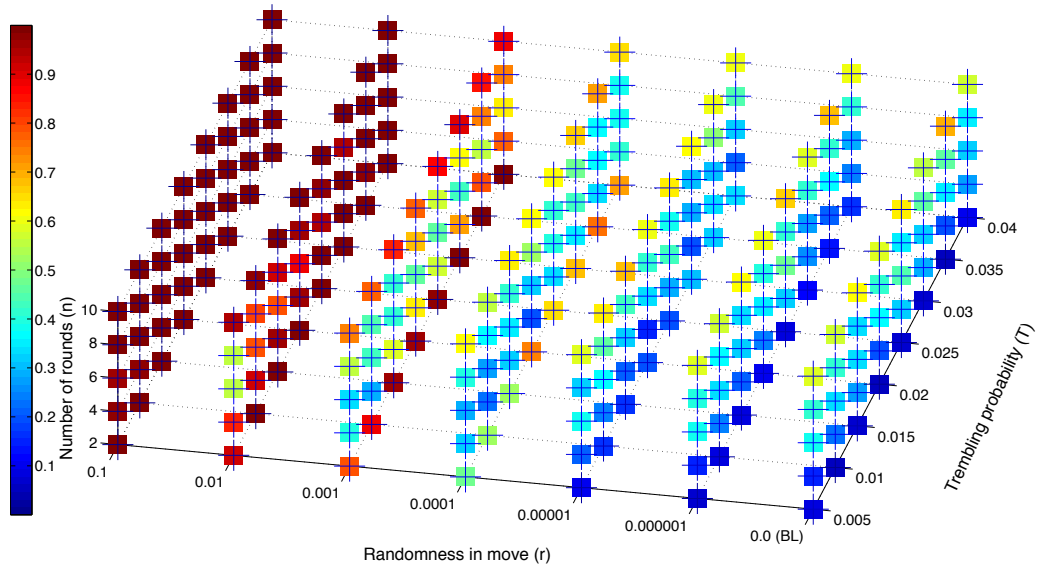


Figure A.24: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure A.69 for the magnitudes of errors in detail.

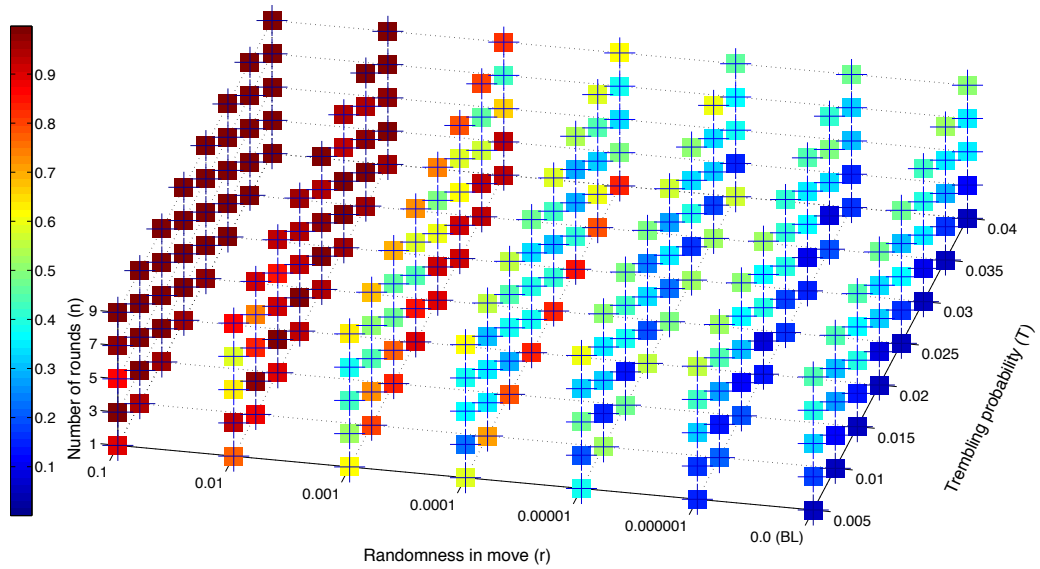


Figure A.25: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure A.70 for the magnitudes of errors in detail.

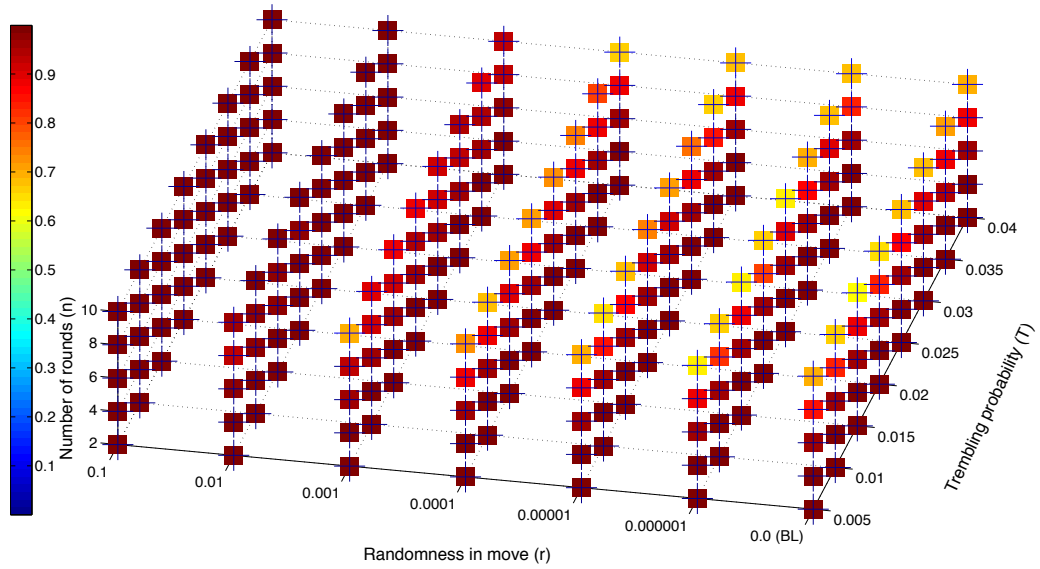


Figure A.26: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure A.71 for the magnitudes of errors in detail.

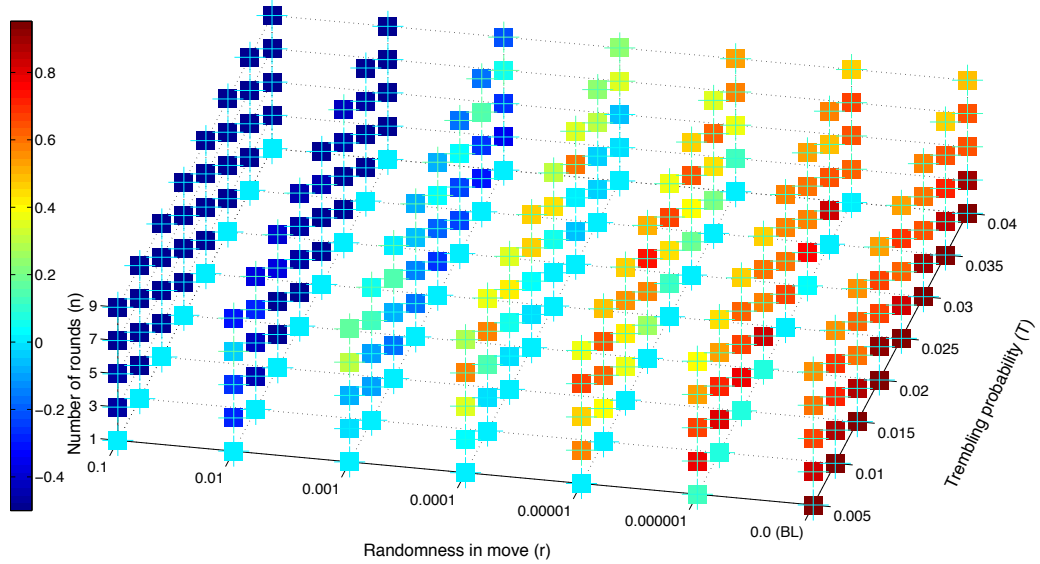


Figure A.27: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure A.72 for the magnitudes of errors in detail.

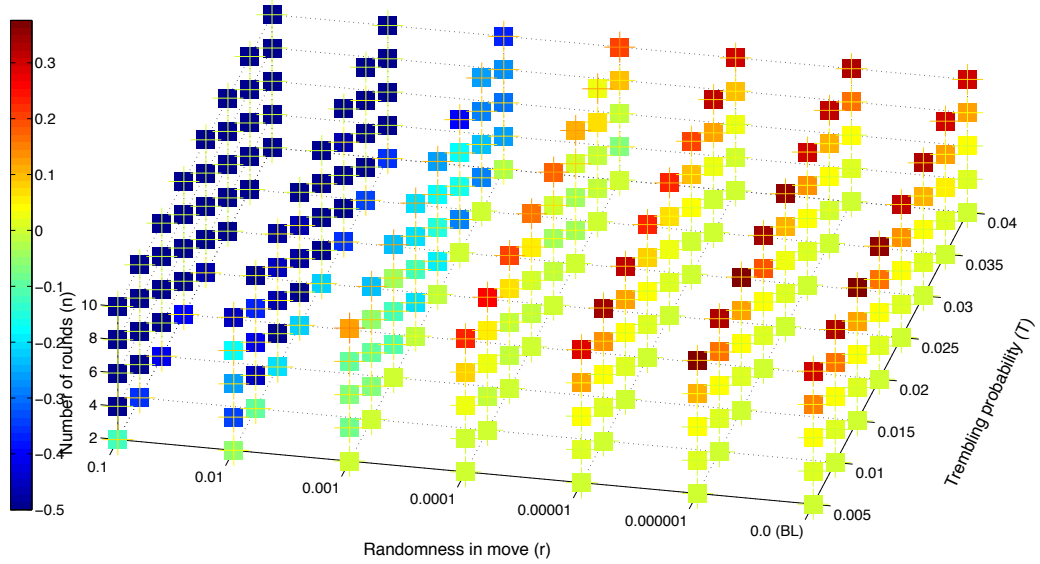


Figure A.28: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure A.73 for the magnitudes of errors in detail.

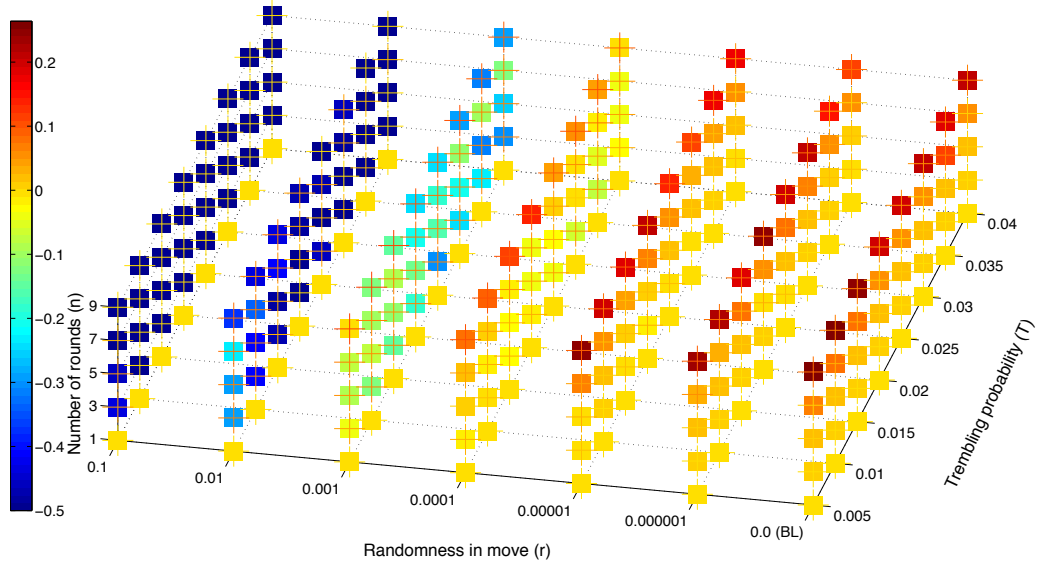


Figure A.29: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure A.74 for the magnitudes of errors in detail.

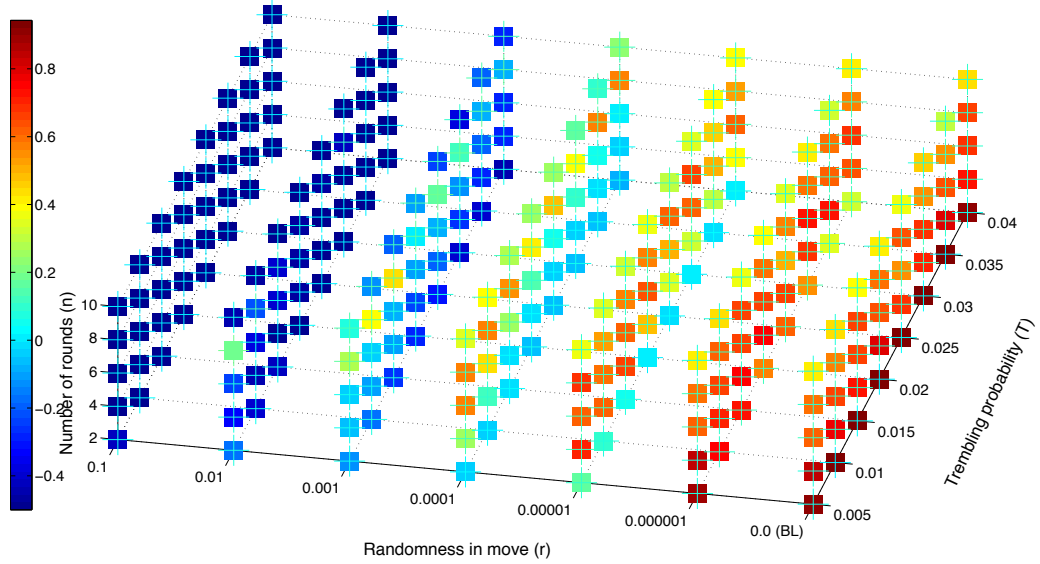


Figure A.30: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure A.75 for the magnitudes of errors in detail.

A.2.3 Implementation and Perception Errors

Average first hitting time

Figure A.31 shows the *average first hitting time*, across 50 runs for all values of r, T and n, for implementation and perception errors.

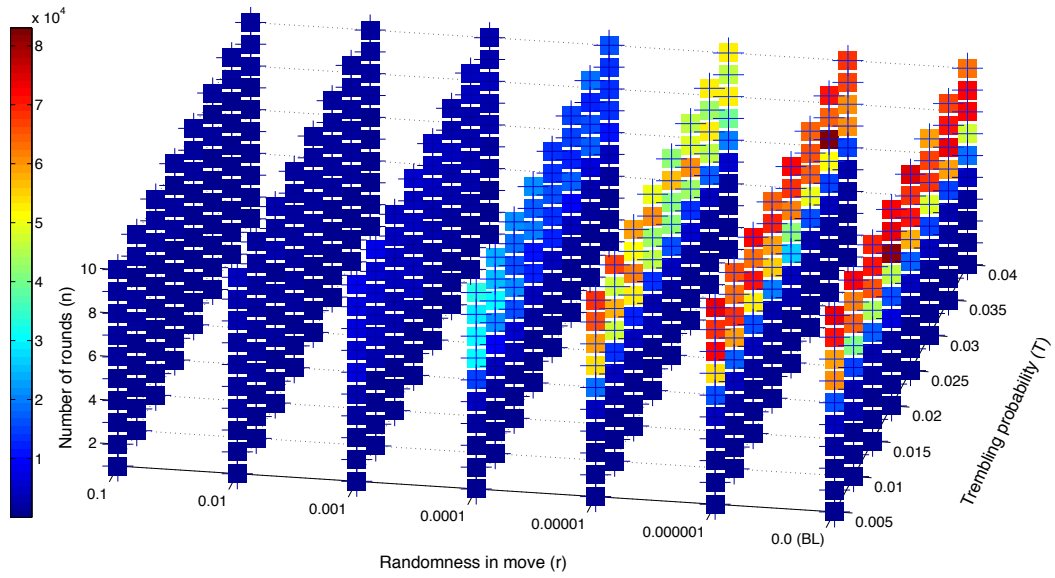


Figure A.31: *Average first hitting time*, across 50 runs for all values of r, T and n (implementation and perception errors). See Figure A.76 for the magnitudes of errors in detail.

Average stay in equilibrium

Figure A.32 shows the *average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n, for implementation and perception errors.

Average distance from equilibrium

Figure A.33 shows the *average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for implementation and perception errors.

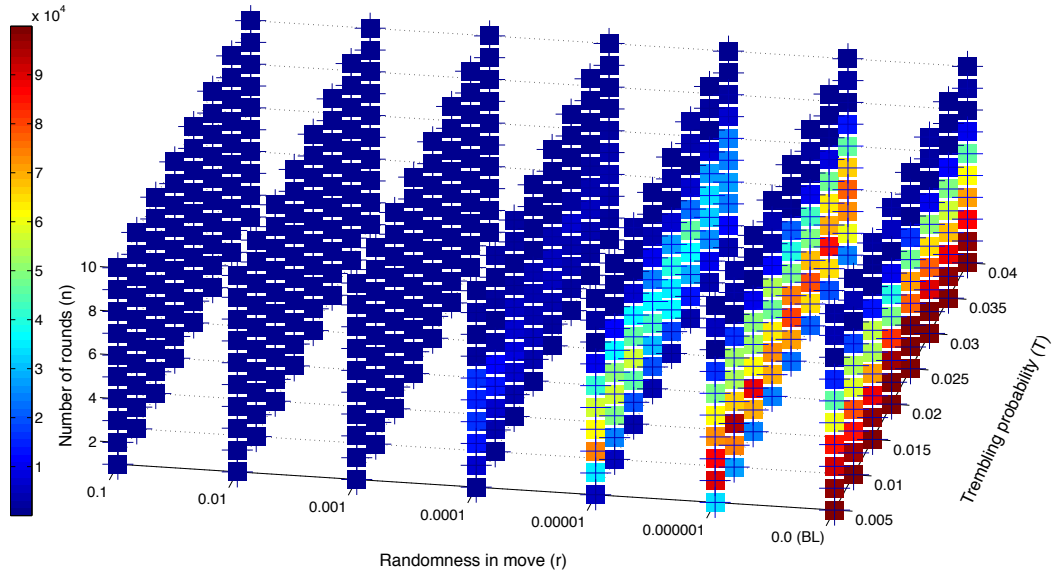


Figure A.32: *Average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n (implementation and perception errors). See Figure A.77 for the magnitudes of errors in detail.

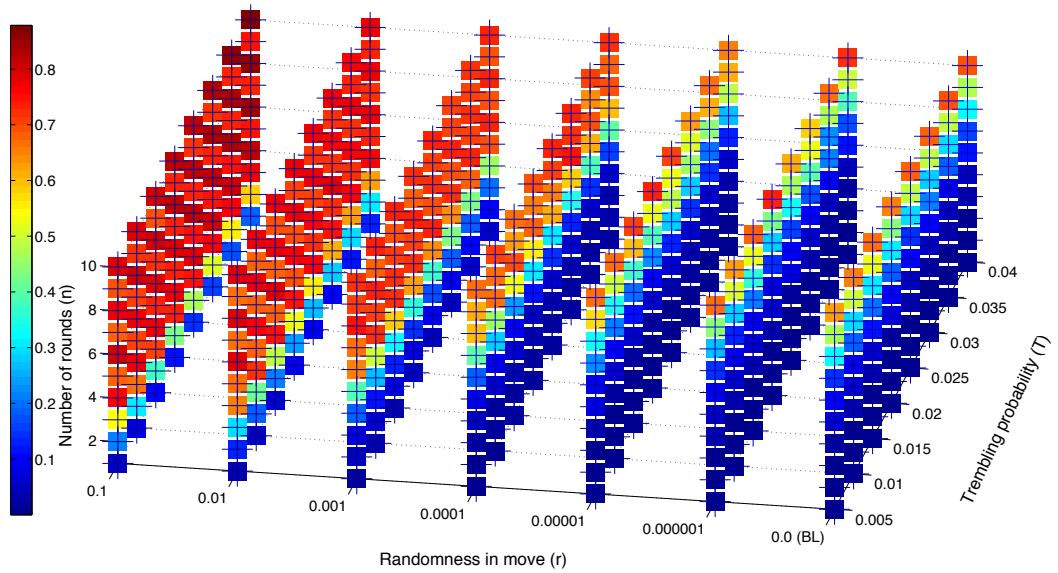


Figure A.33: *Average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (implementation and perception errors). See Figure A.78 for the magnitudes of errors in detail.

Average payoff

Figures A.34 and A.35 show the *average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

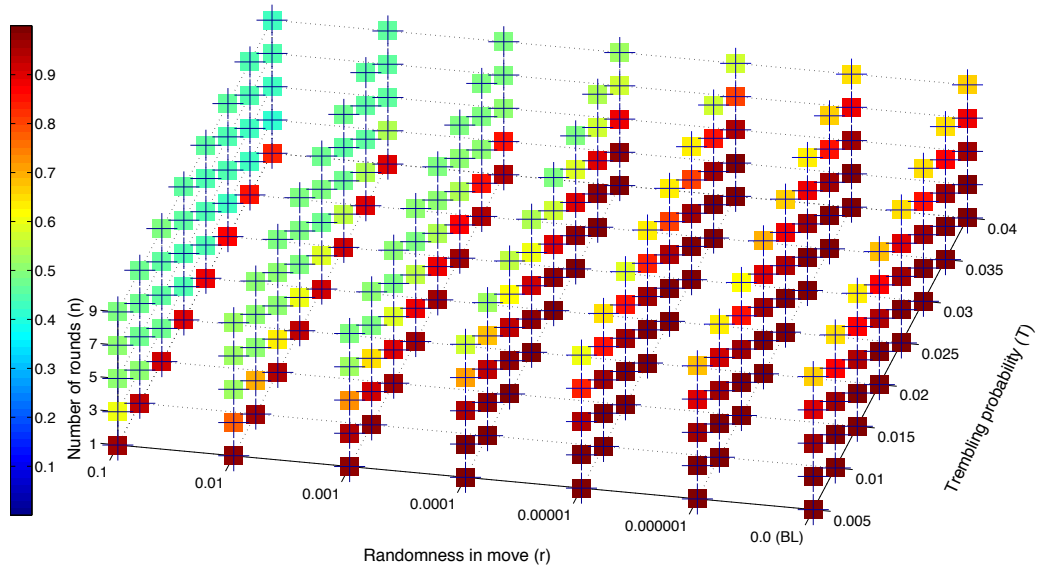


Figure A.34: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure A.79 for the magnitudes of errors in detail.

Figures A.36 and A.37 show the *average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Average maximum payoff

Figures A.38 and A.39 show the *average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Figures A.40 and A.41 show the *average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

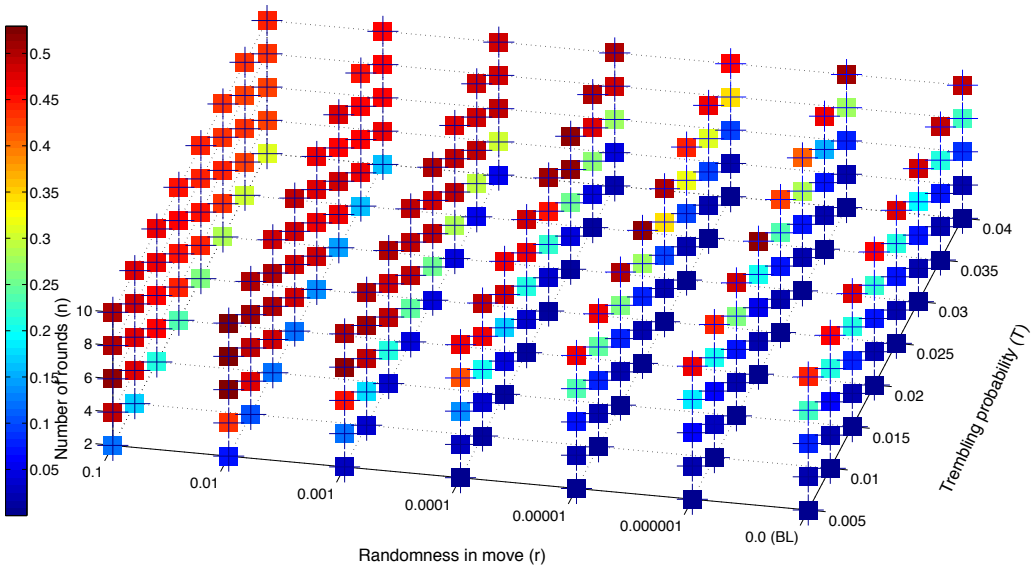


Figure A.35: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure A.80 for the magnitudes of errors in detail.

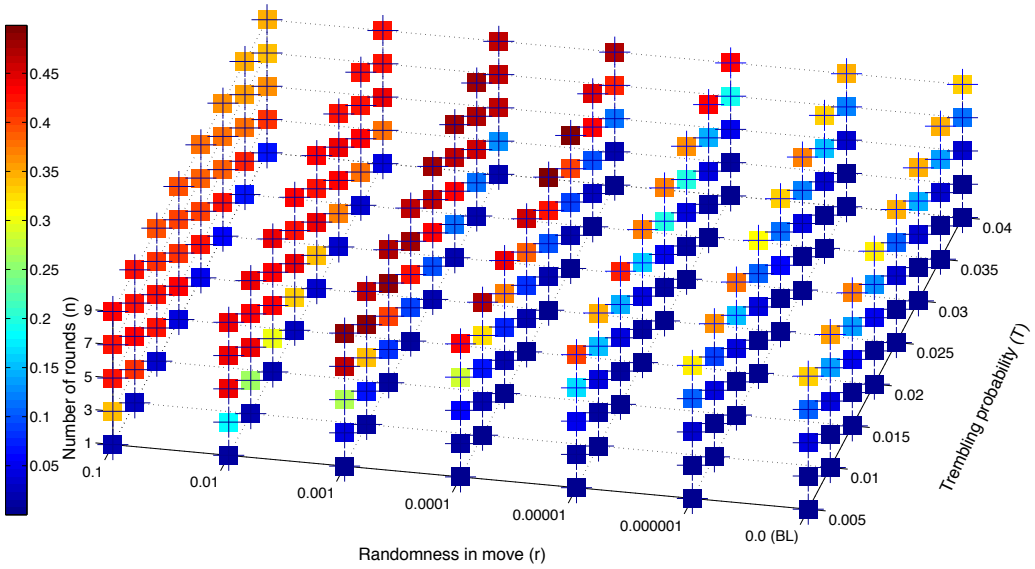


Figure A.36: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure A.81 for the magnitudes of errors in detail.

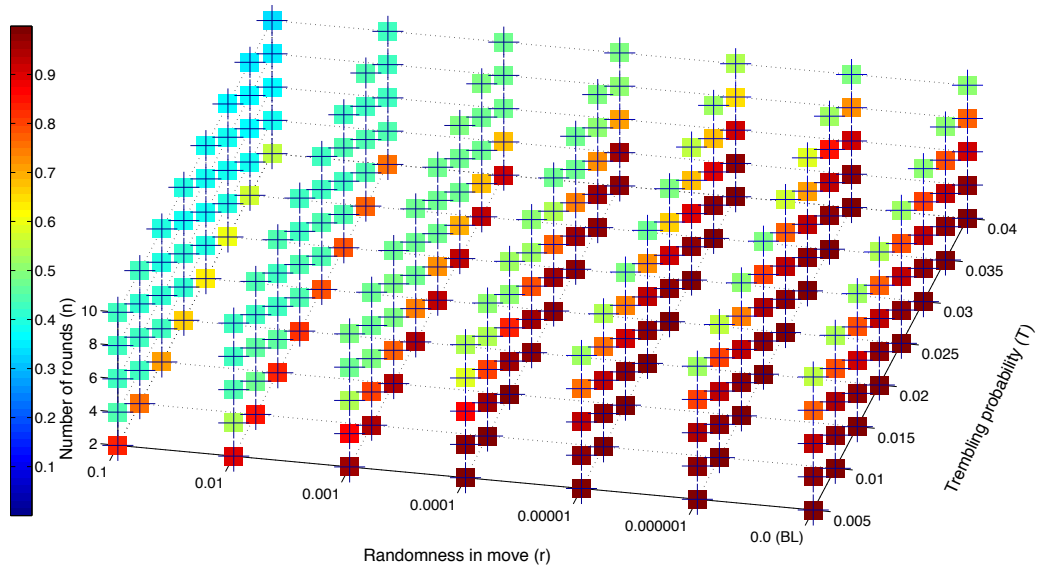


Figure A.37: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure A.82 for the magnitudes of errors in detail.

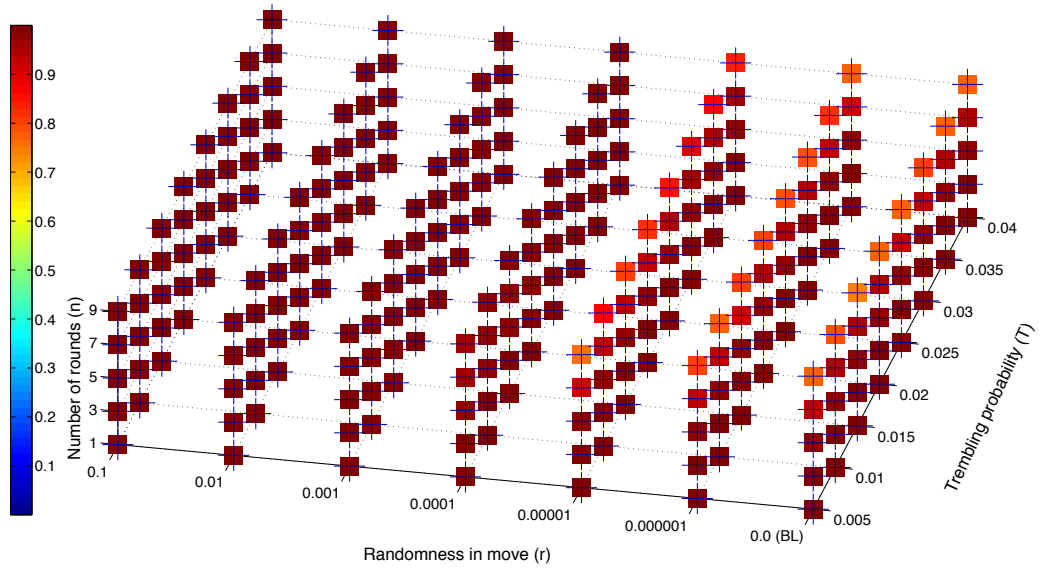


Figure A.38: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure A.83 for the magnitudes of errors in detail.

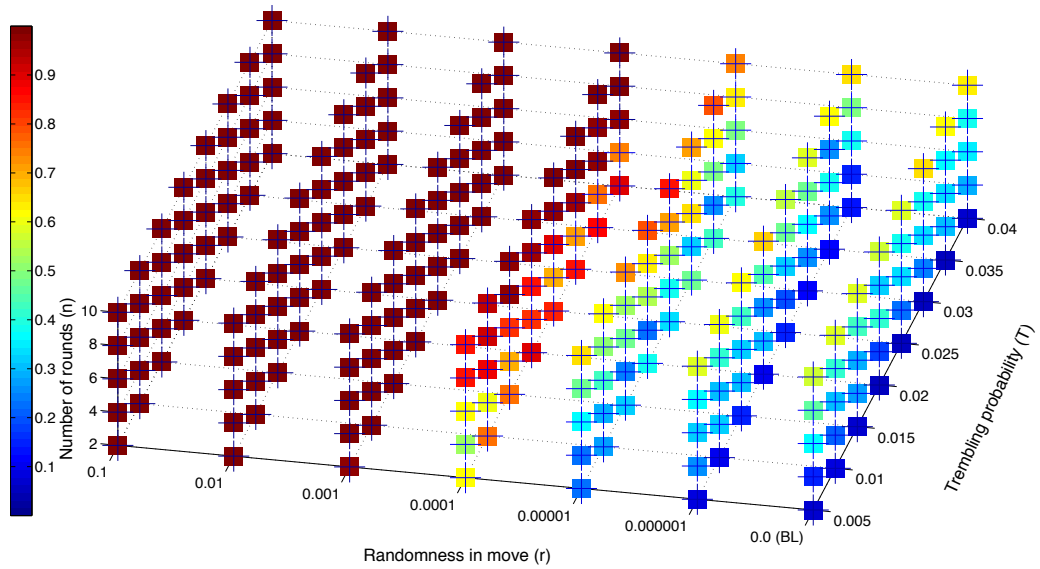


Figure A.39: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure A.84 for the magnitudes of errors in detail.

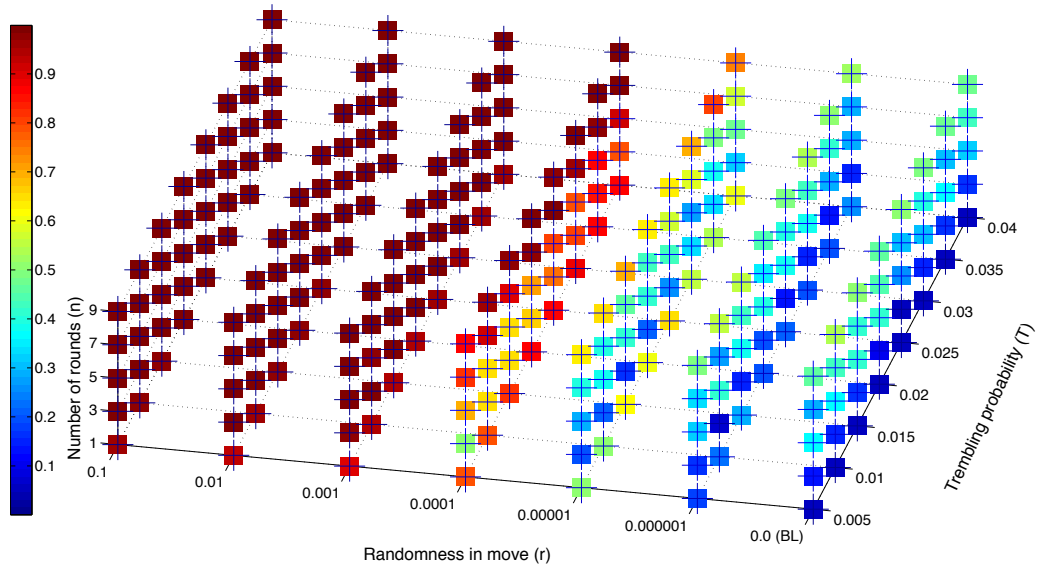


Figure A.40: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure A.85 for the magnitudes of errors in detail.

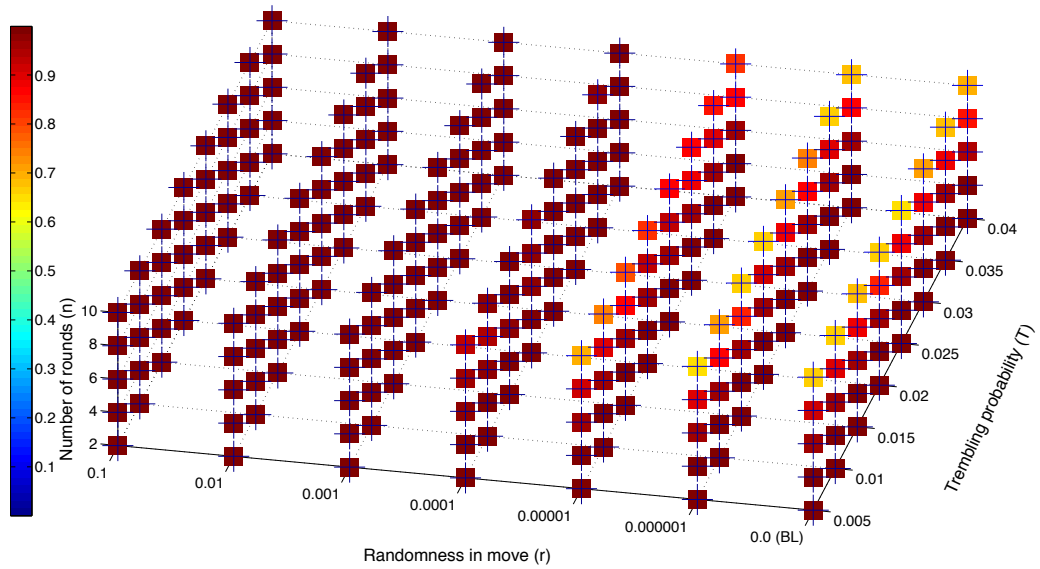


Figure A.41: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure A.86 for the magnitudes of errors in detail.

Average minimum payoff

Figures A.42 and A.43 show the *average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Figures A.44 and A.45 show the *average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

A.3 Confidence Across 50 Runs

For each type of bounded rationality, we plot the standard error (95% confidence) for each evaluation metric, for each game and rationality bound setting. This is in the same order as the graphs in Section A.2 for easy reference.

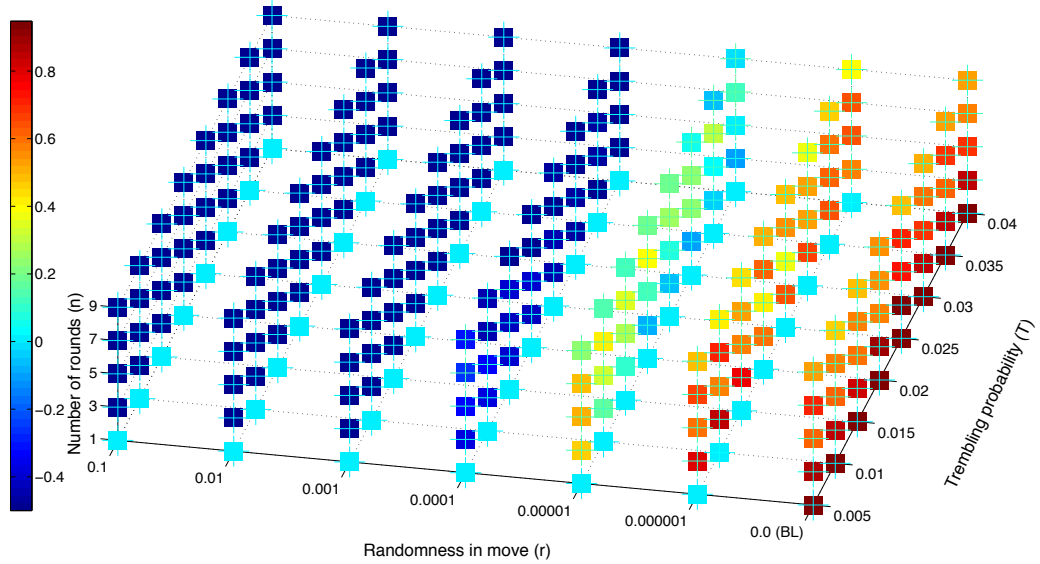


Figure A.42: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure A.87 for the magnitudes of errors in detail.

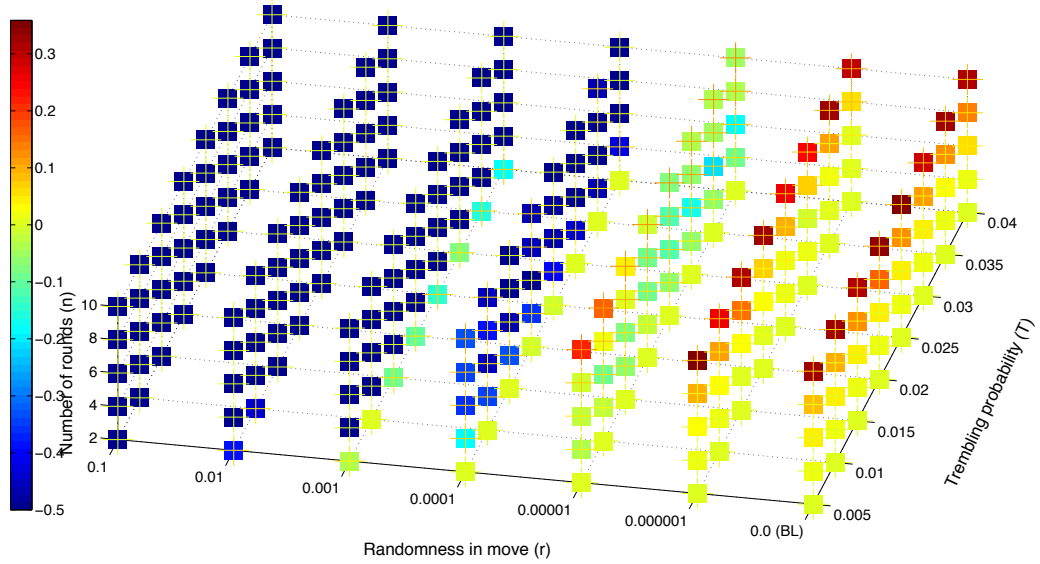


Figure A.43: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure A.88 for the magnitudes of errors in detail.

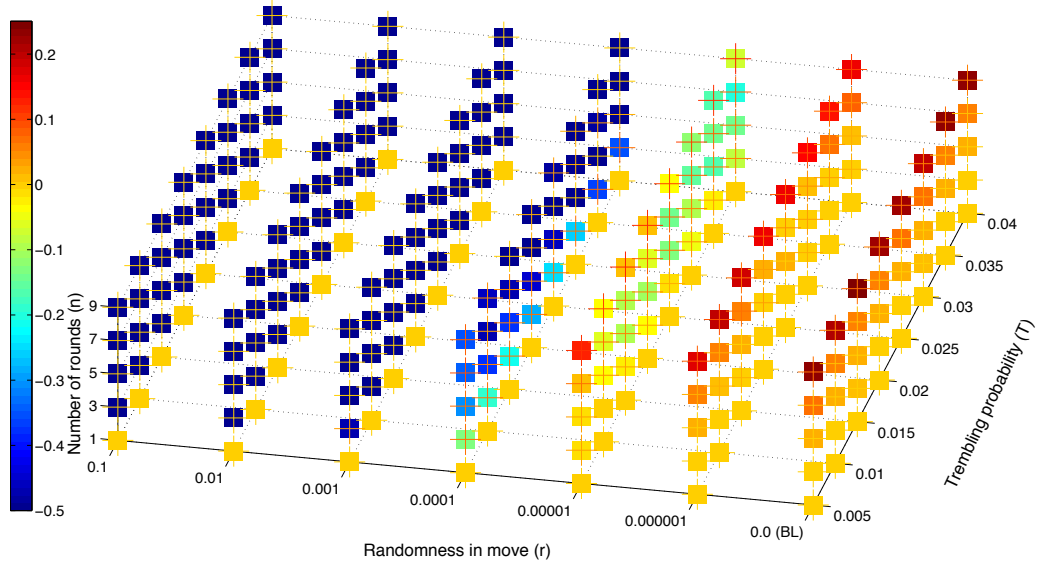


Figure A.44: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure A.89 for the magnitudes of errors in detail.

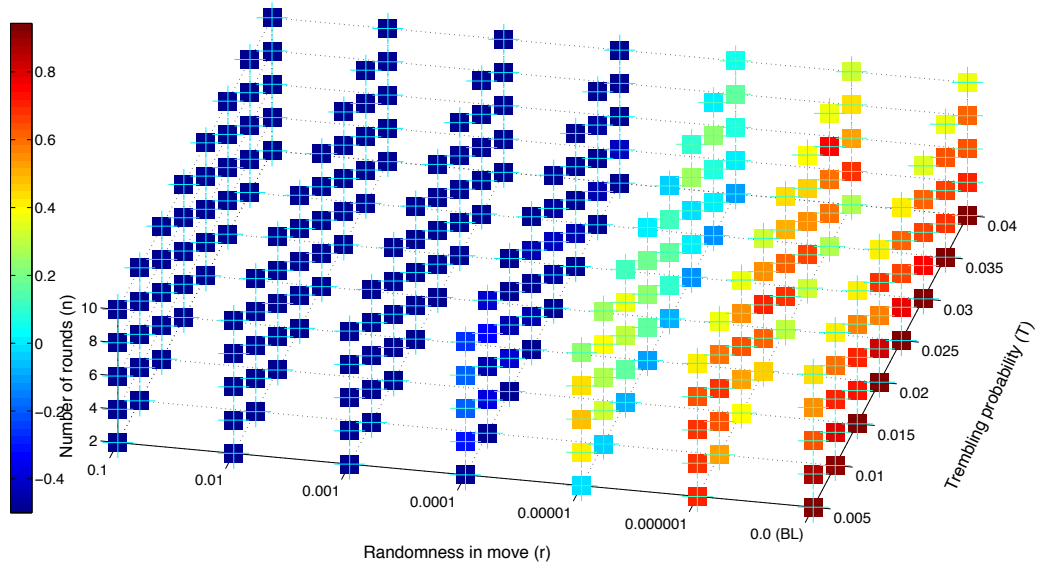


Figure A.45: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure A.90 for the magnitudes of errors in detail.

A.3.1 Implementation Errors

Error at 95% confidence for average first hitting time

Figure A.46 shows the *error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n, for implementation errors.

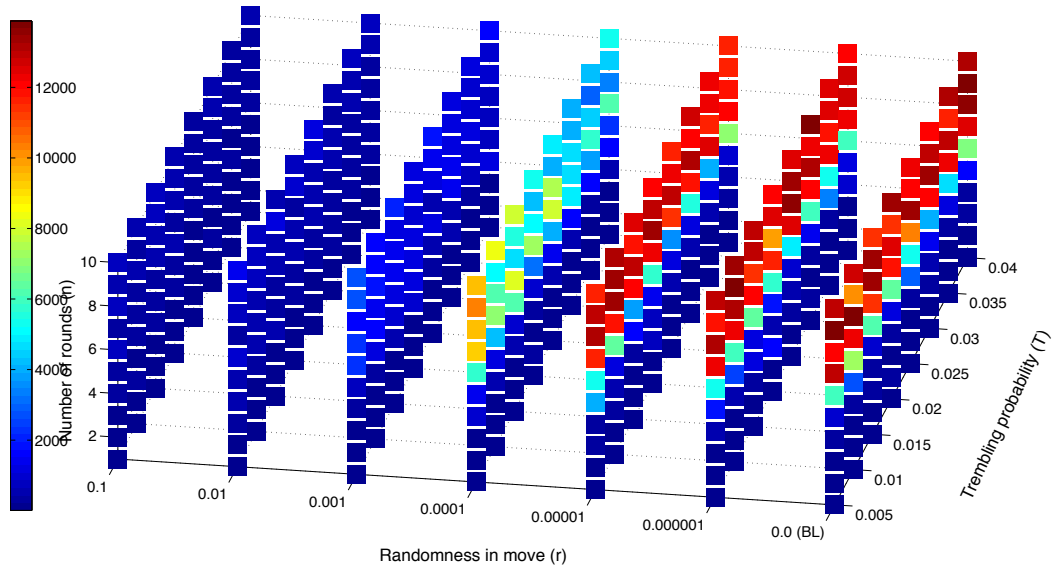


Figure A.46: *Error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n (implementation errors).

Error at 95% confidence for average stay in equilibrium

Figure A.47 shows the *error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n, for implementation errors.

Error at 95% confidence for average distance from equilibrium

Figure A.48 shows the *error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for implementation errors.

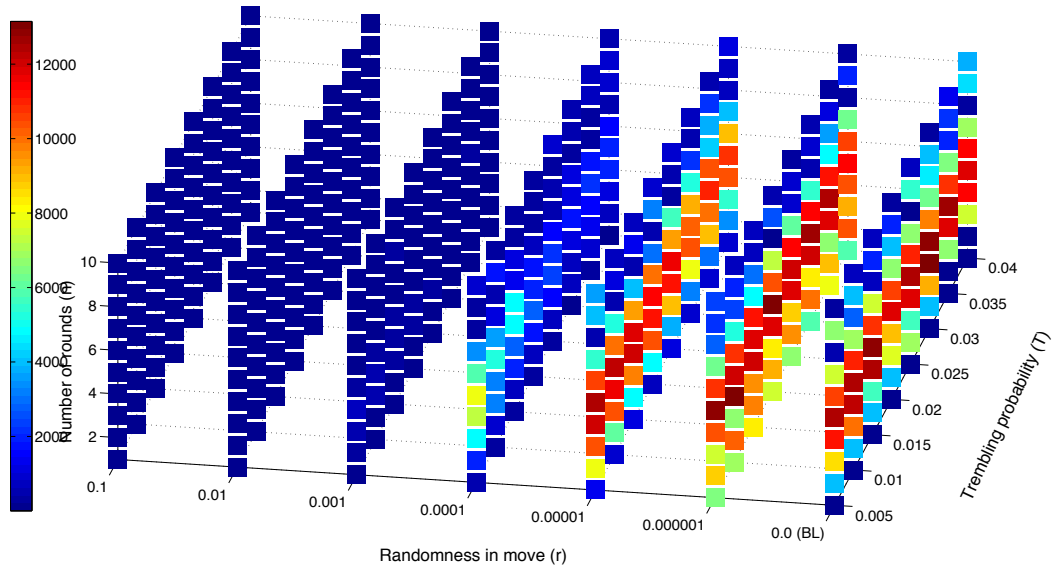


Figure A.47: *Error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n (implementation errors).

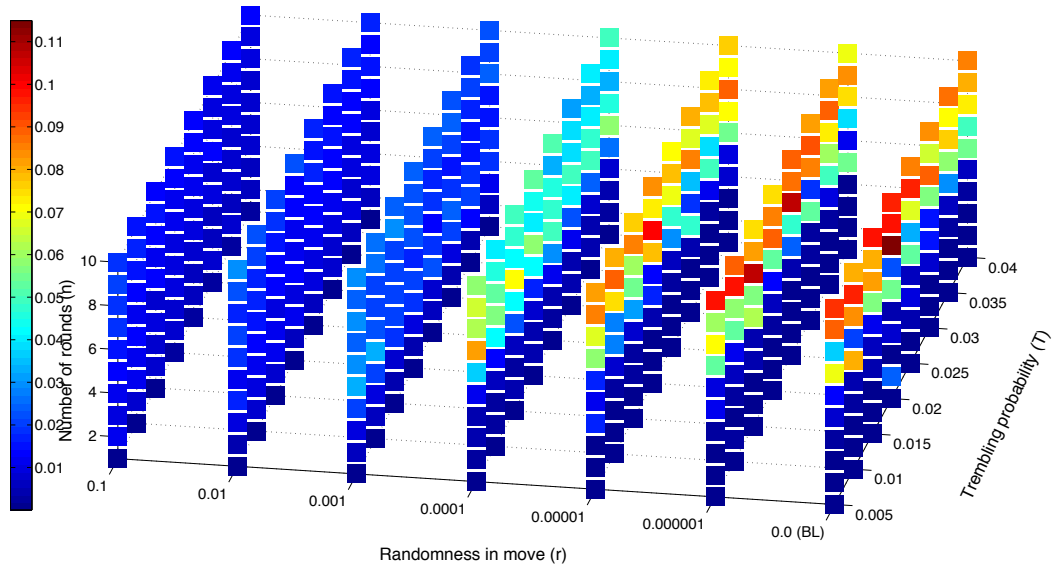


Figure A.48: *Error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (implementation errors).

Error at 95% confidence for average payoff

Figures A.49 and A.50 show the *error at* 95% *confidence for average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

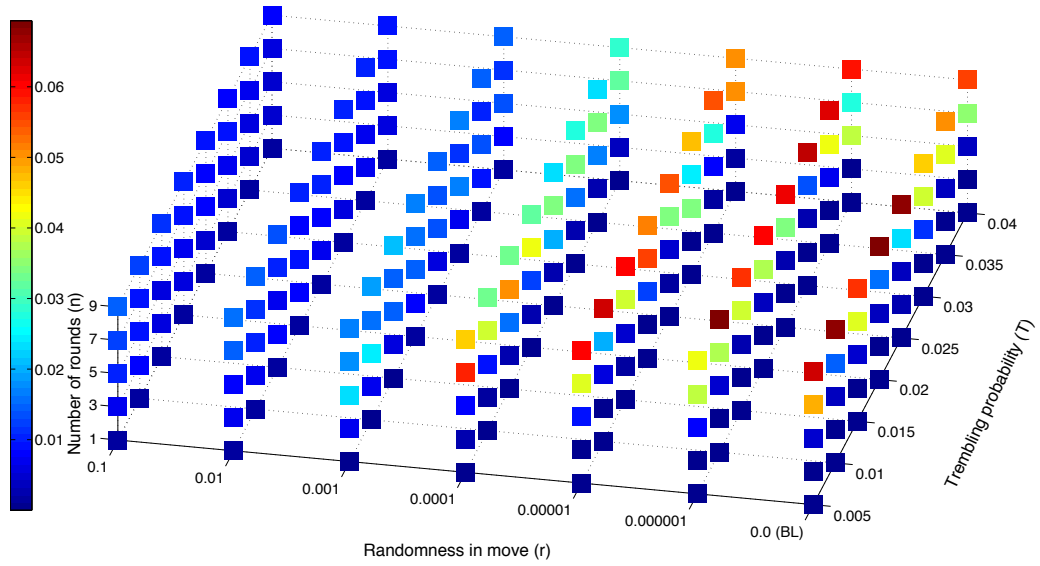


Figure A.49: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

Figures A.51 and A.52 show the *error at* 95% *confidence for average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Error at 95% confidence for average maximum payoff

Figures A.53 and A.54 show the *error at* 95% *confidence for average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Figures A.55 and A.56 show the *error at* 95% *confidence for average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

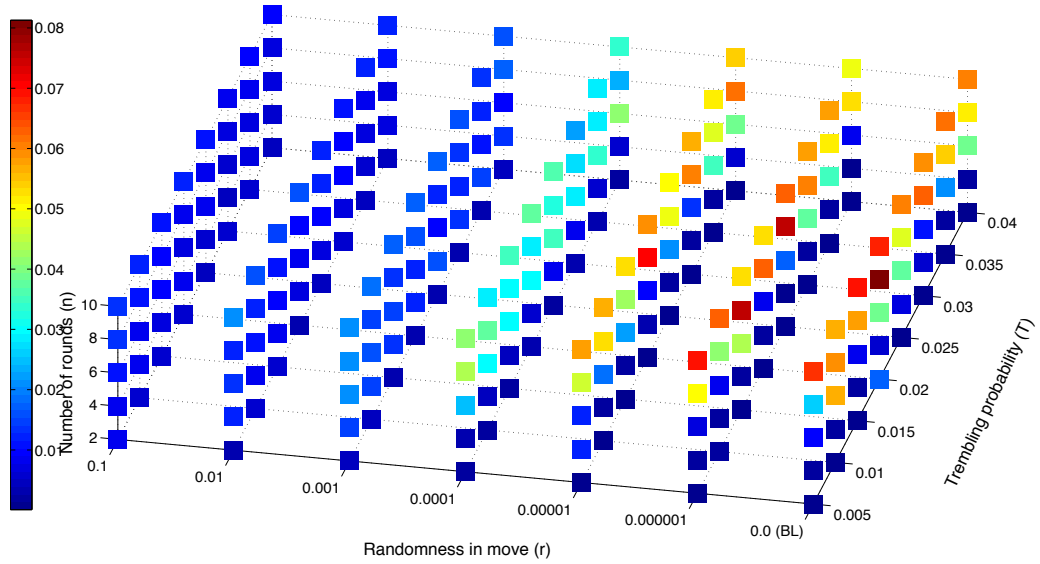


Figure A.50: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

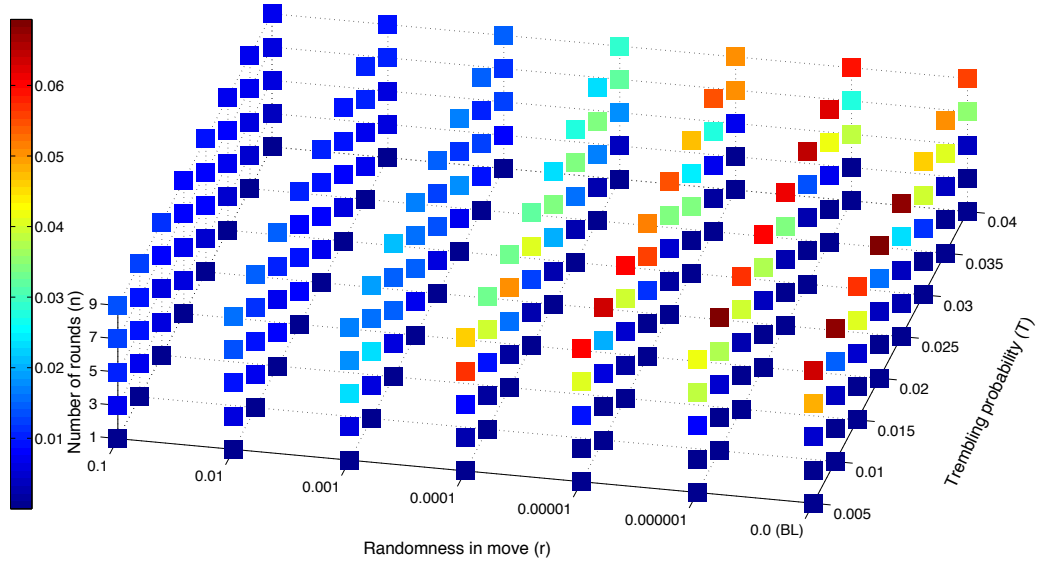


Figure A.51: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

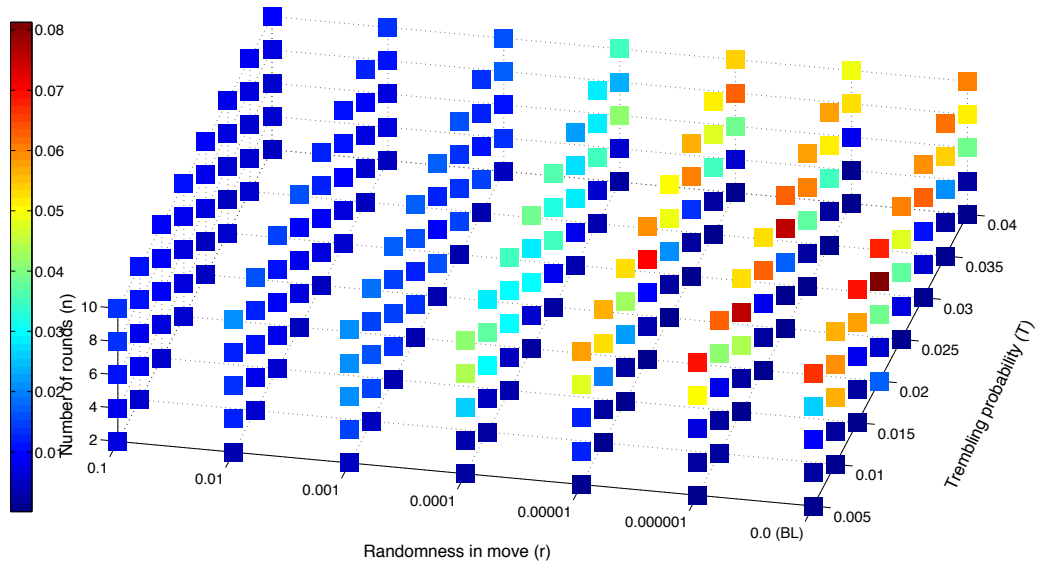


Figure A.52: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

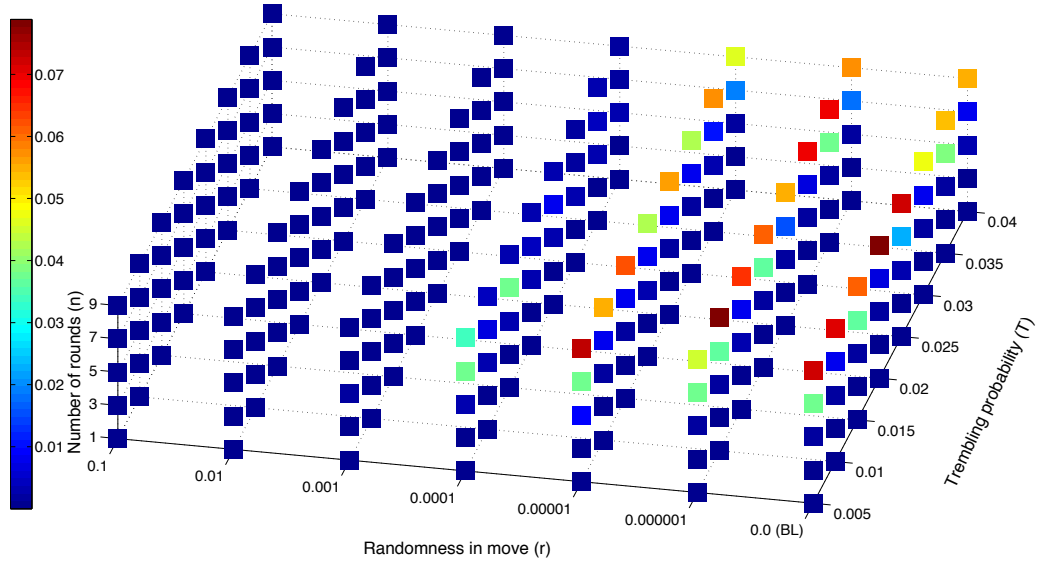


Figure A.53: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

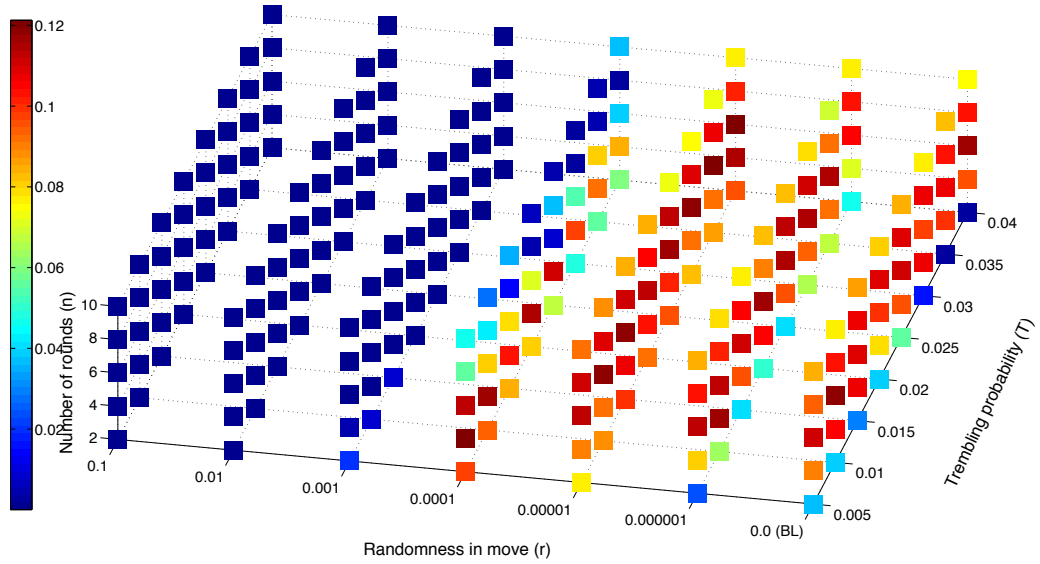


Figure A.54: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

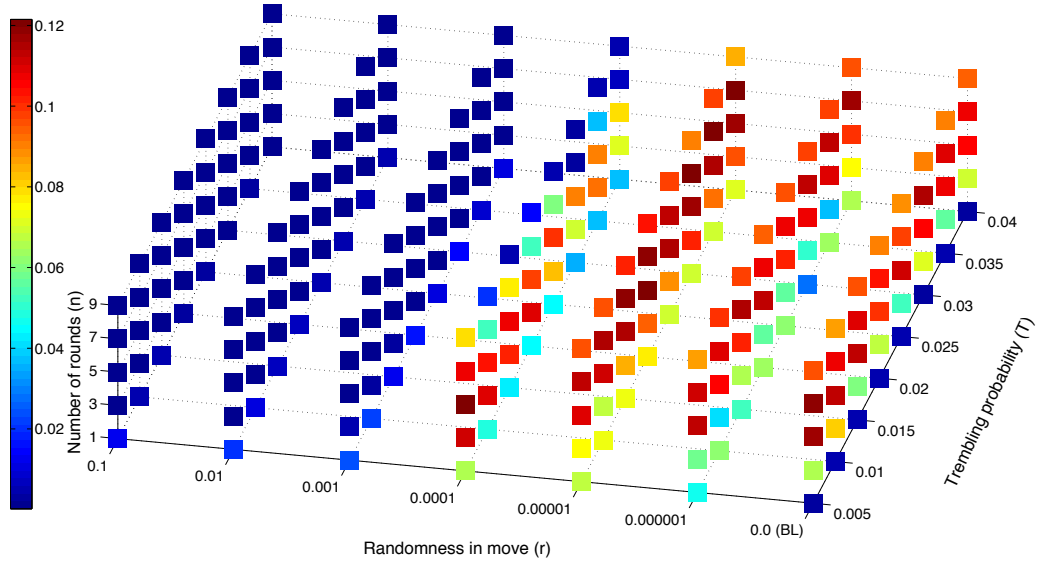


Figure A.55: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

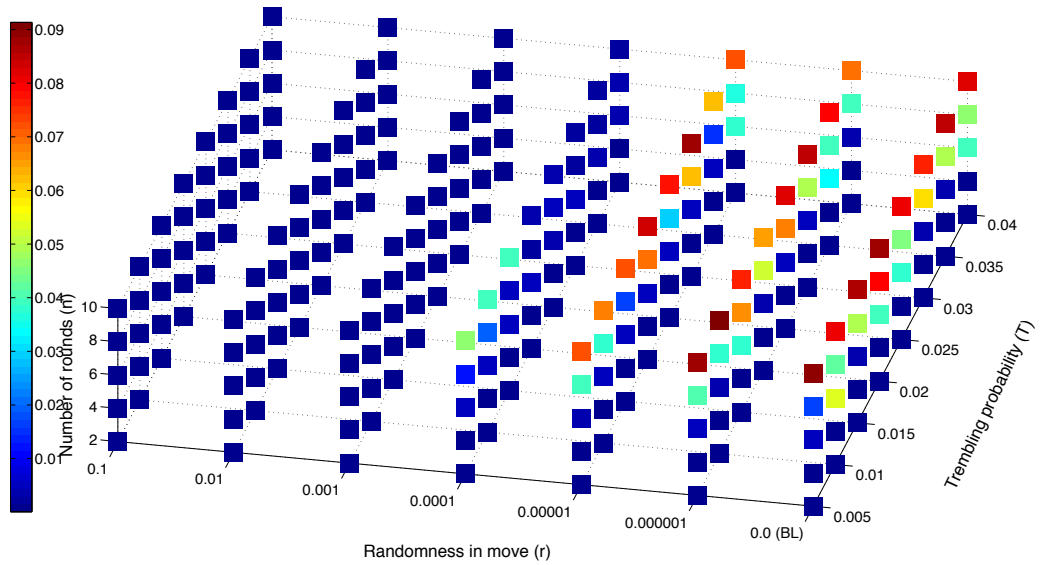


Figure A.56: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

Error at 95% confidence for average minimum payoff

Figures A.57 and A.58 show the *error at* 95% *confidence for average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Figures A.59 and A.60 show the *error at* 95% *confidence for average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

A.3.2 Perception Errors

Error at 95% confidence for average first hitting time

Figure A.61 shows the *error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n, for perception errors.

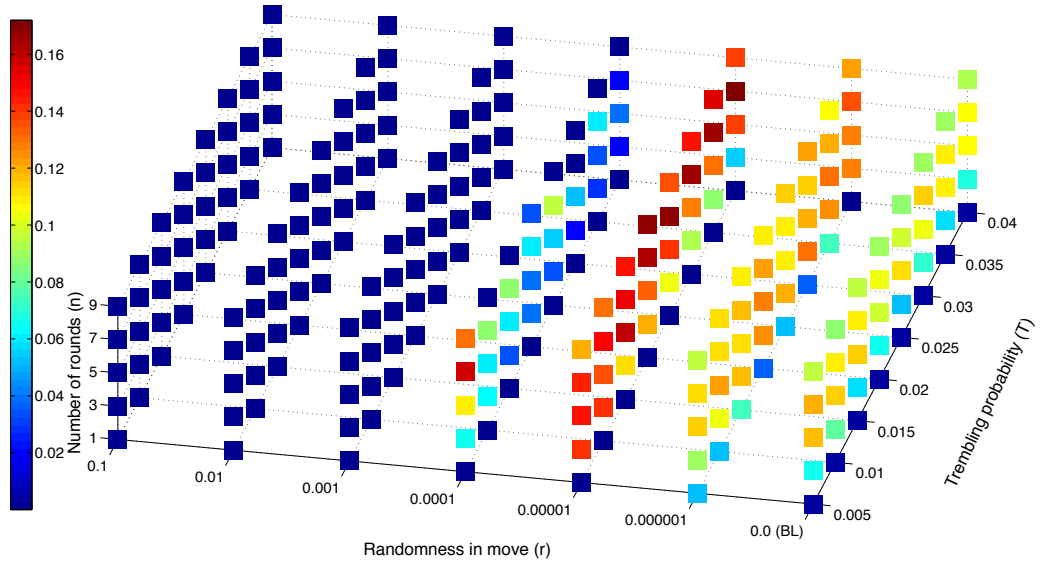


Figure A.57: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

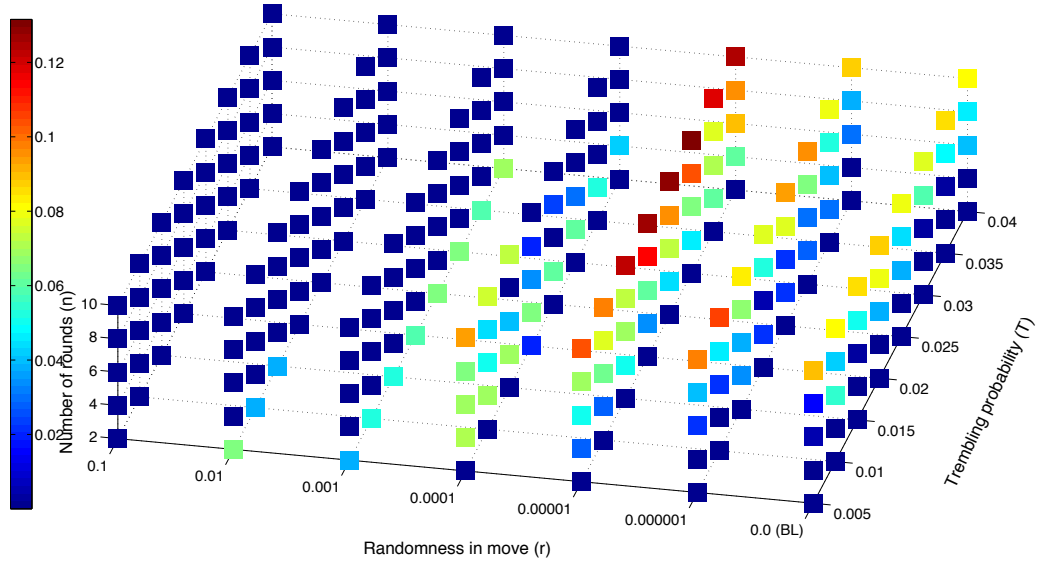


Figure A.58: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

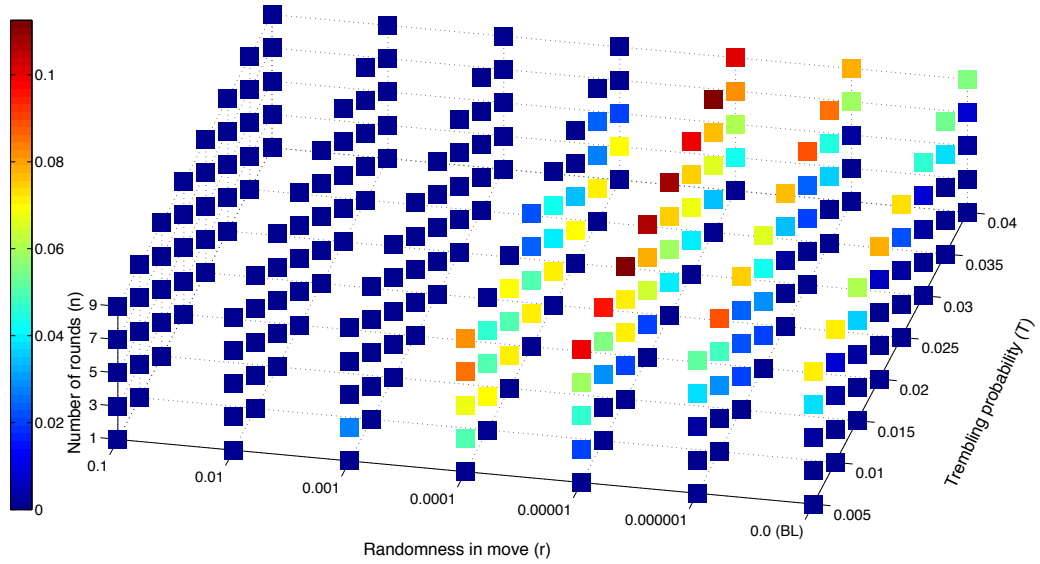


Figure A.59: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

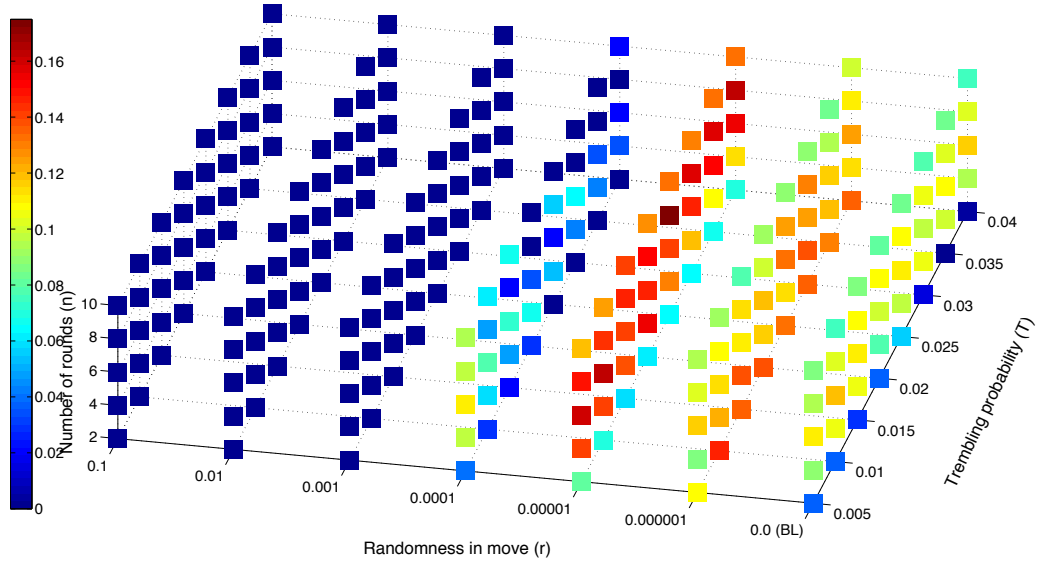


Figure A.60: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

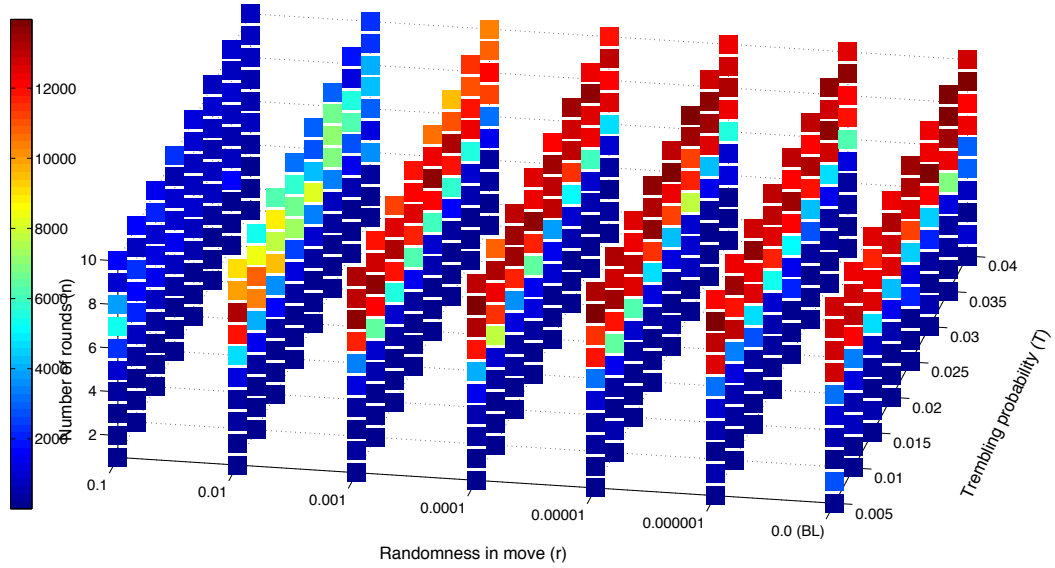


Figure A.61: *Error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n (perception errors).

Error at 95% confidence for average stay in equilibrium

Figure A.62 shows the *error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n, for perception errors.

Error at 95% confidence for average distance from equilibrium

Figure A.63 shows the *error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for perception errors.

Error at 95% confidence for average payoff

Figures A.64 and A.65 show the *error at* 95% *confidence for average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures A.66 and A.67 show the *error at* 95% *confidence for average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being

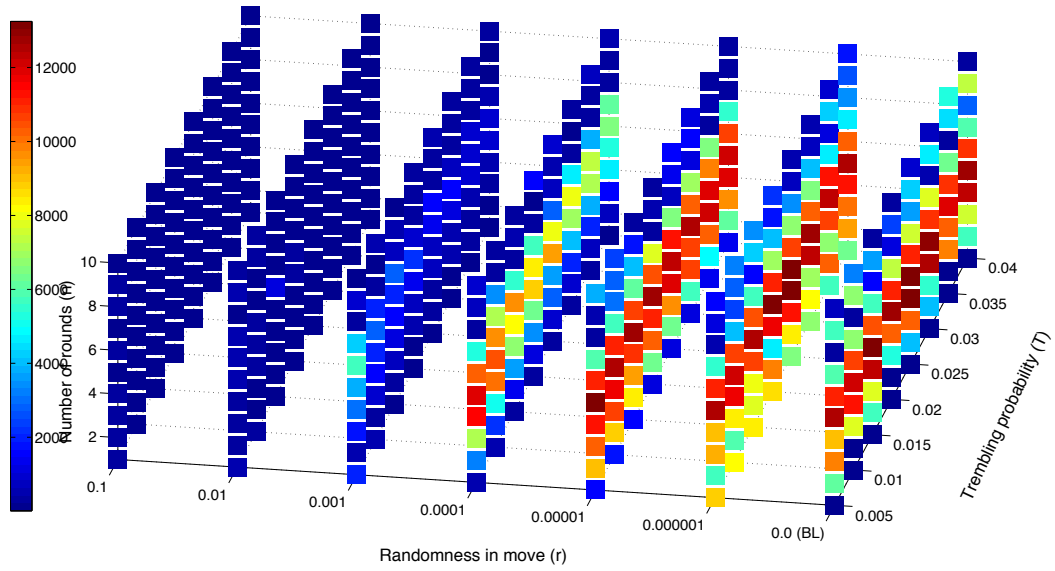


Figure A.62: *Error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n (perception errors).

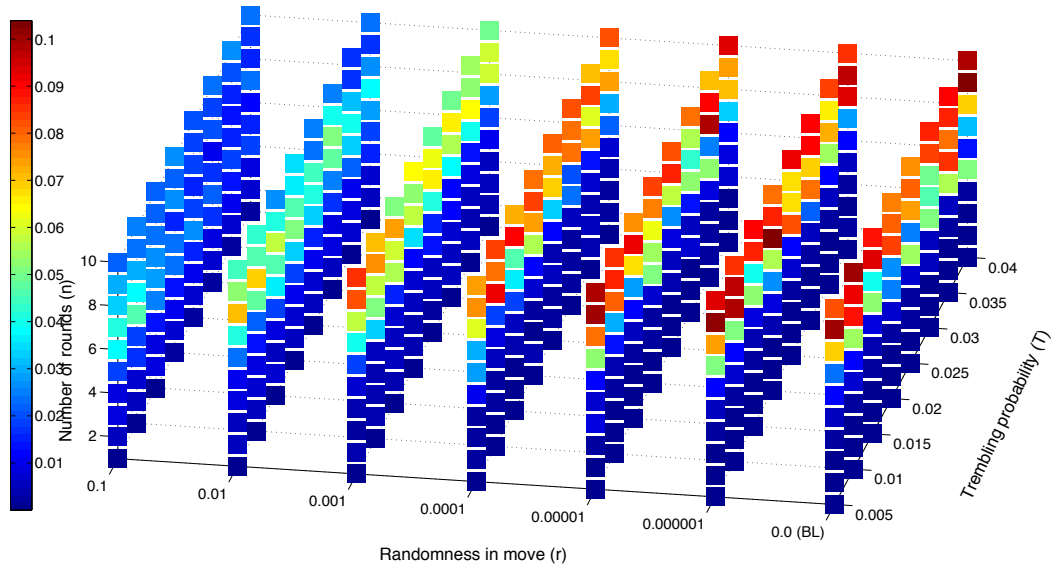


Figure A.63: *Error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (perception errors).

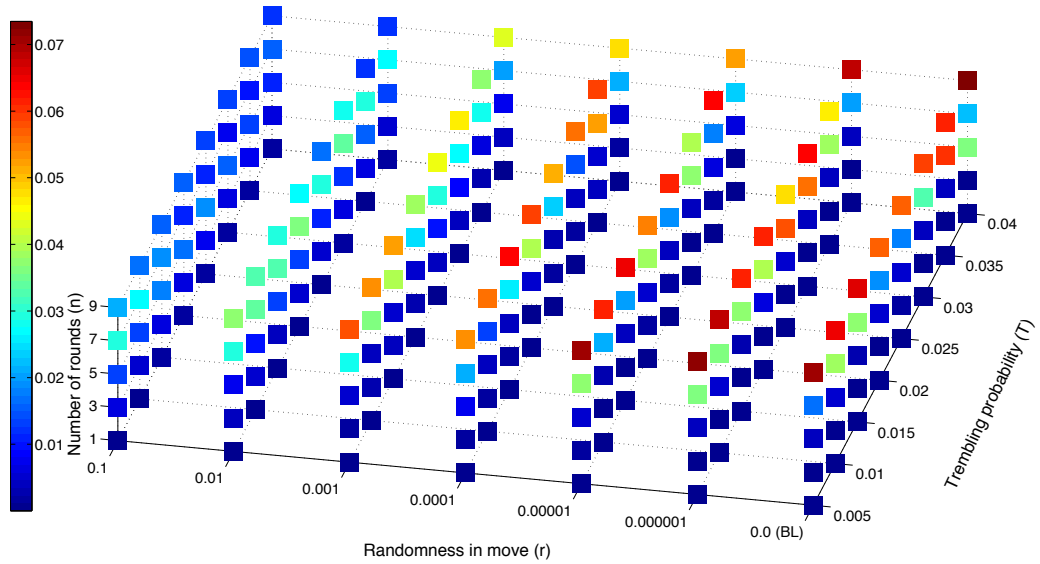


Figure A.64: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

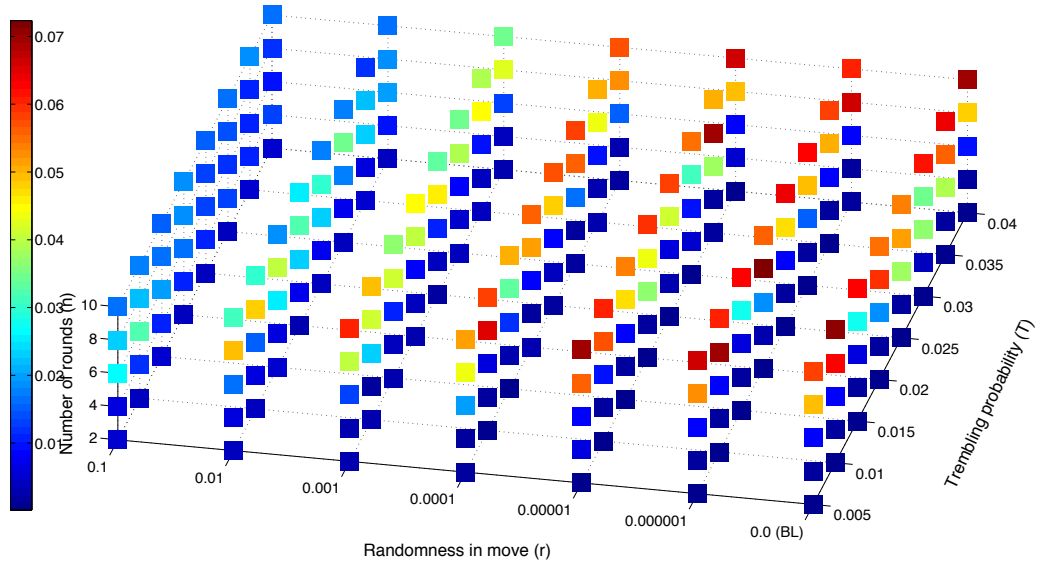


Figure A.65: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

odd and even respectively, for perception errors.

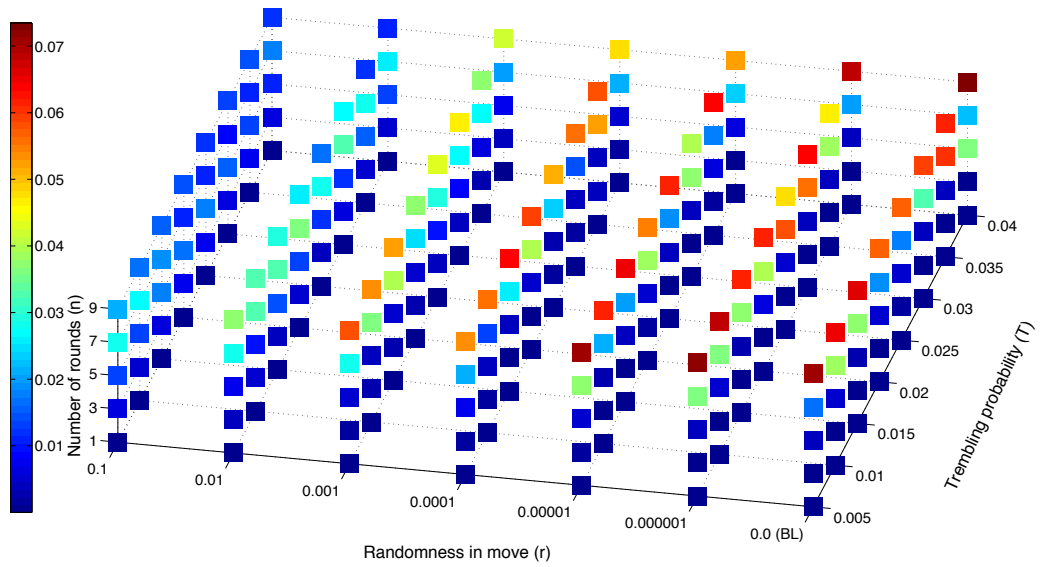


Figure A.66: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

Error at 95% confidence for average maximum payoff

Figures A.68 and A.69 show the *error at* 95% *confidence for average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures A.70 and A.71 show the *error at* 95% *confidence for average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Error at 95% confidence for average minimum payoff

Figures A.72 and A.73 show the *error at* 95% *confidence for average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures A.74 and A.75 show the *error at* 95% *confidence for average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

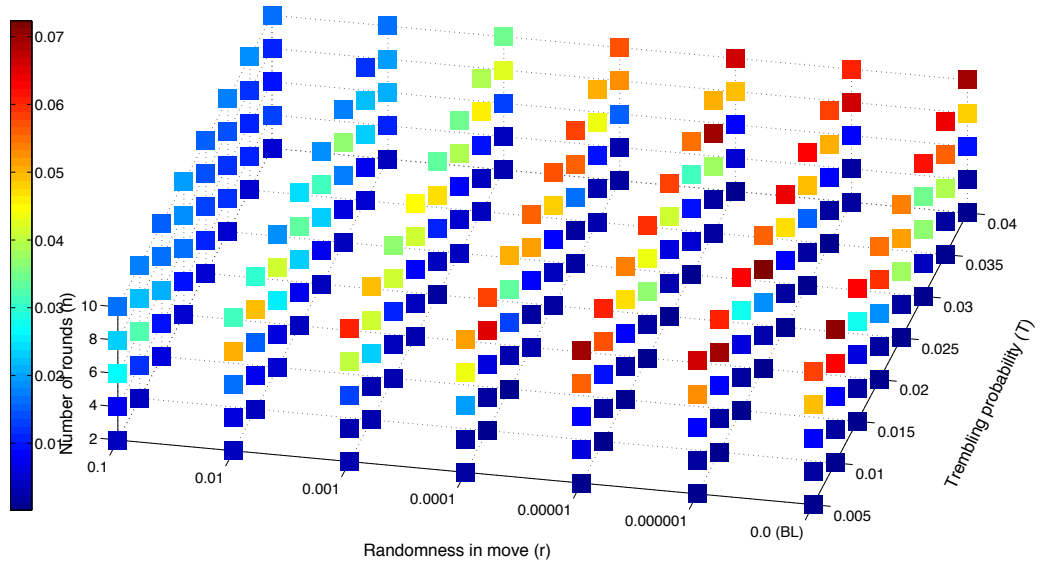


Figure A.67: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

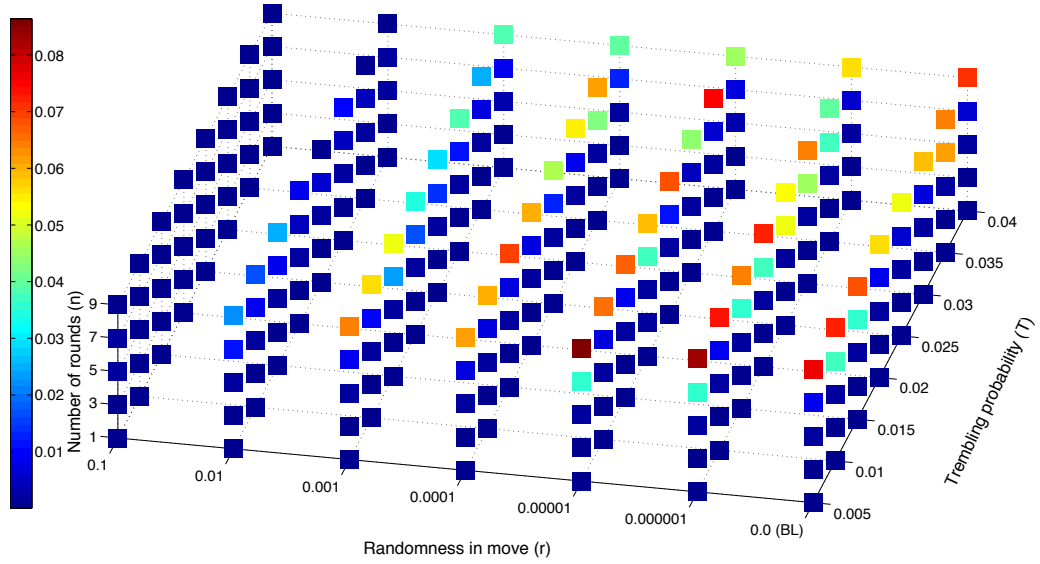


Figure A.68: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

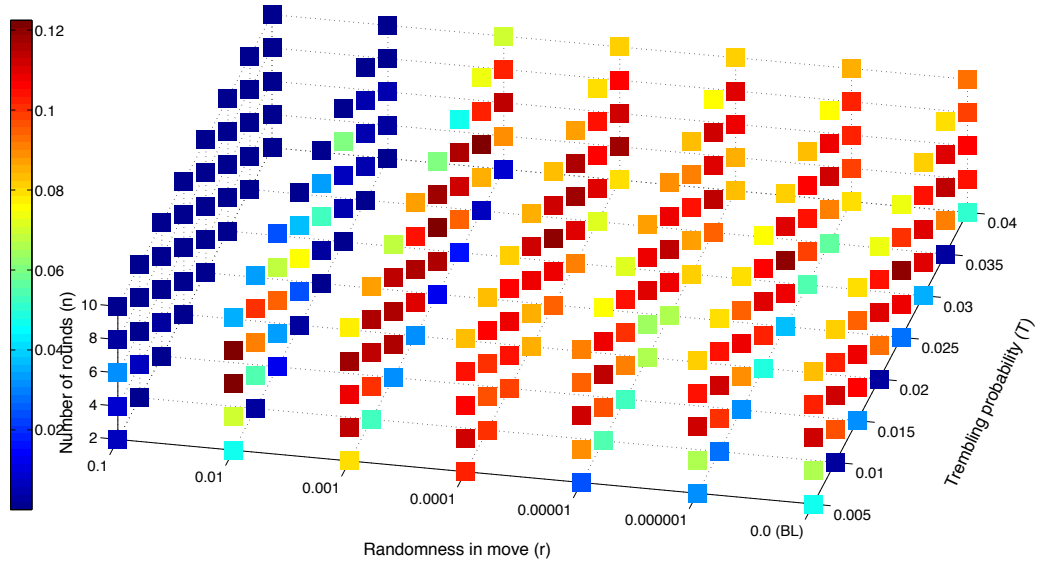


Figure A.69: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

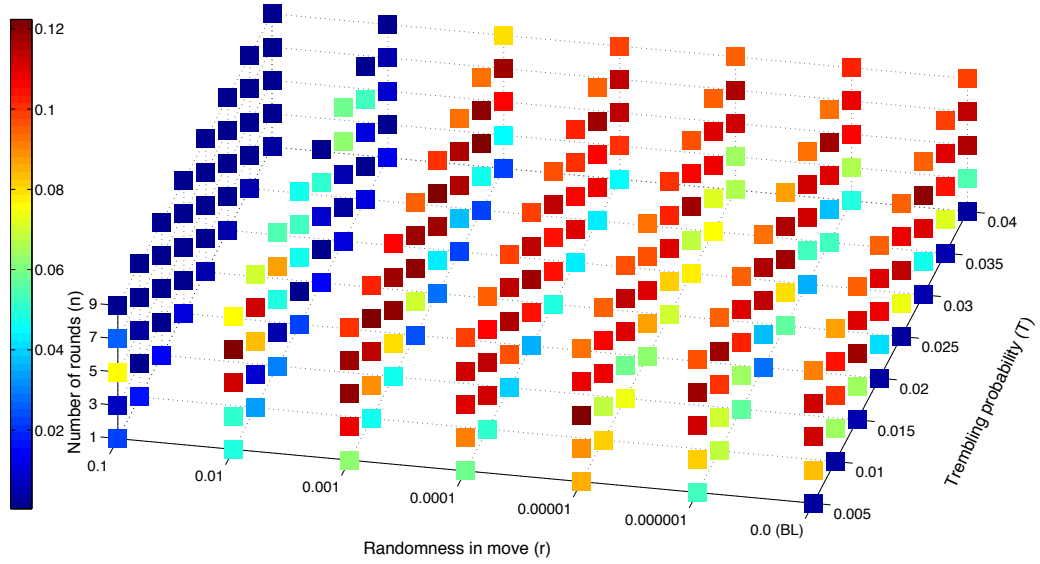


Figure A.70: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

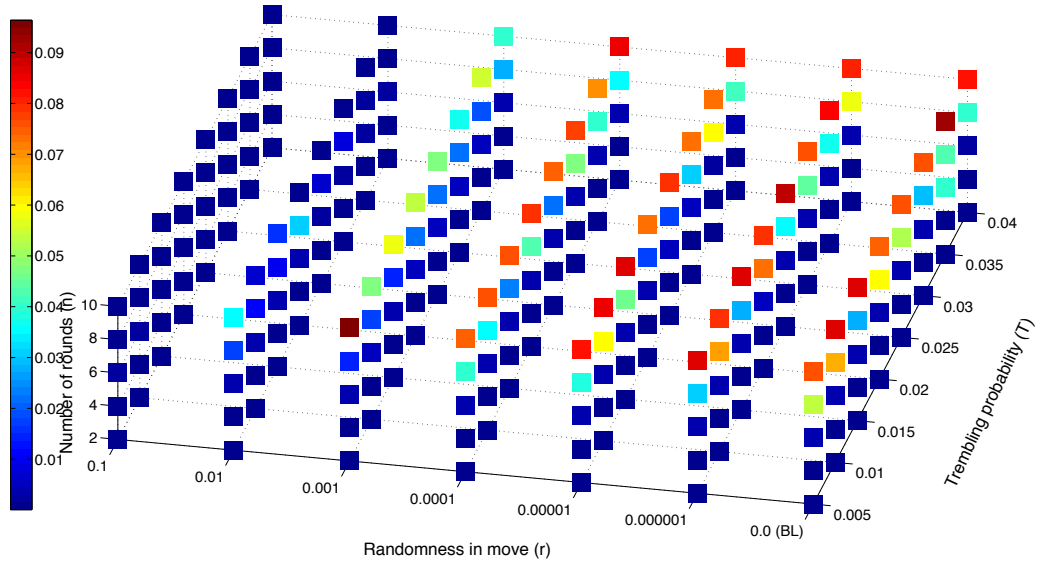


Figure A.71: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

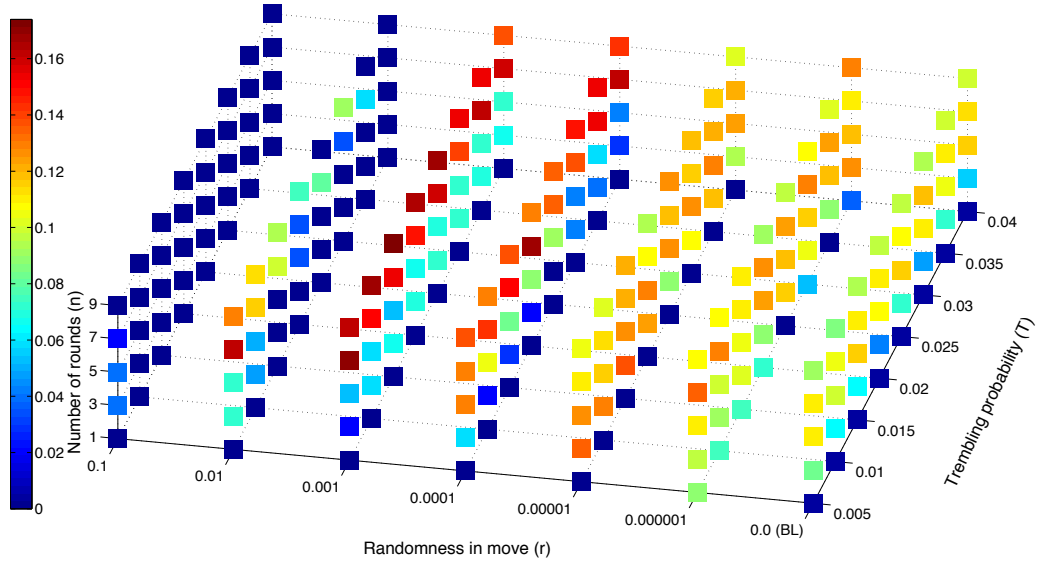


Figure A.72: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

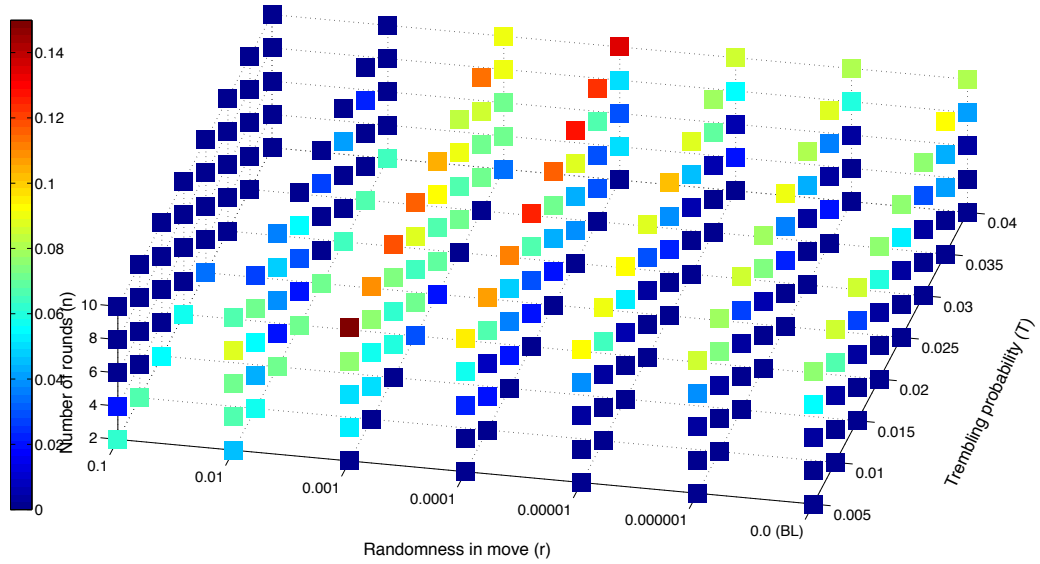


Figure A.73: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

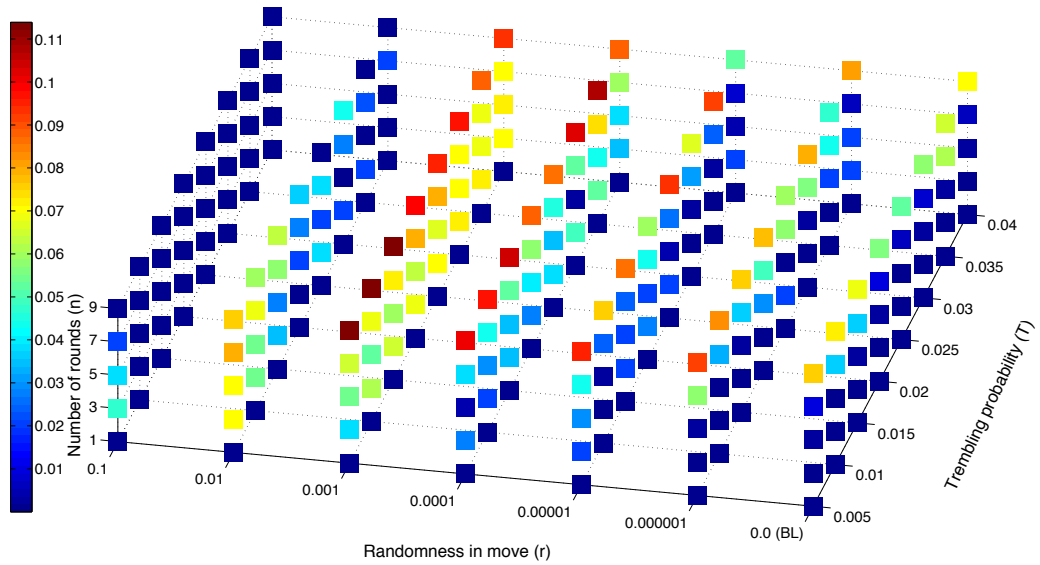


Figure A.74: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

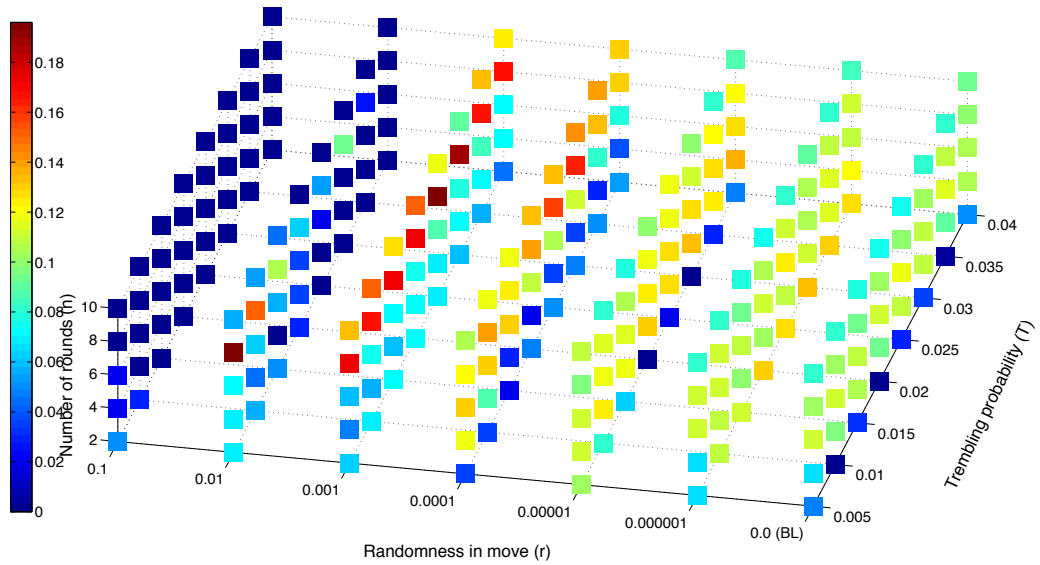


Figure A.75: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

A.3.3 Implementation and Perception Errors

Error at 95% confidence for average first hitting time

Figure A.76 shows the *error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n, for implementation and perception errors.

Error at 95% confidence for average stay in equilibrium

Figure A.77 shows the *error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n, for implementation and perception errors.

Error at 95% confidence for average distance from equilibrium

Figure A.78 shows the *error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for implementation and perception errors.

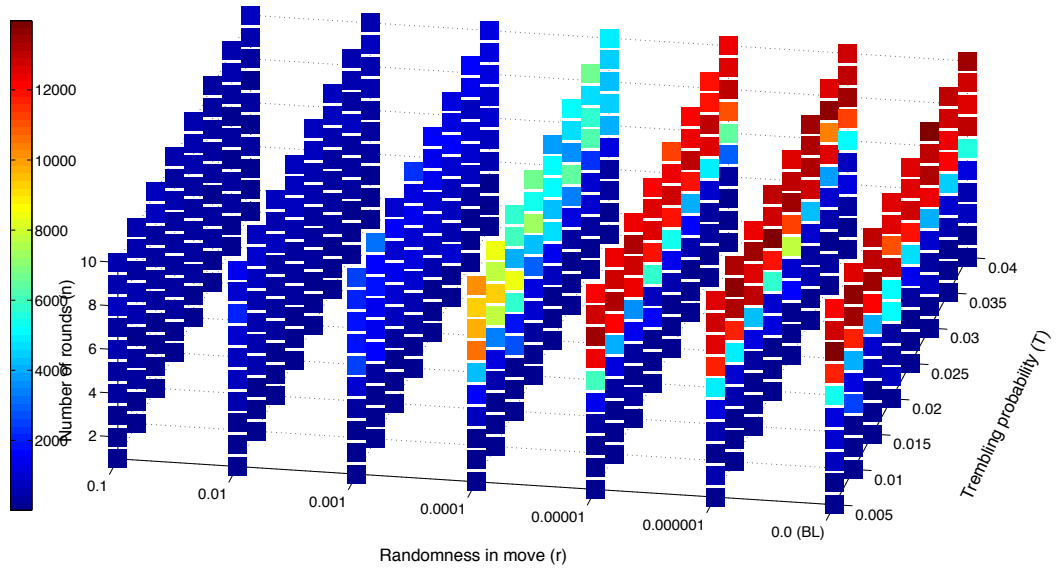


Figure A.76: *Error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n (implementation and perception errors).

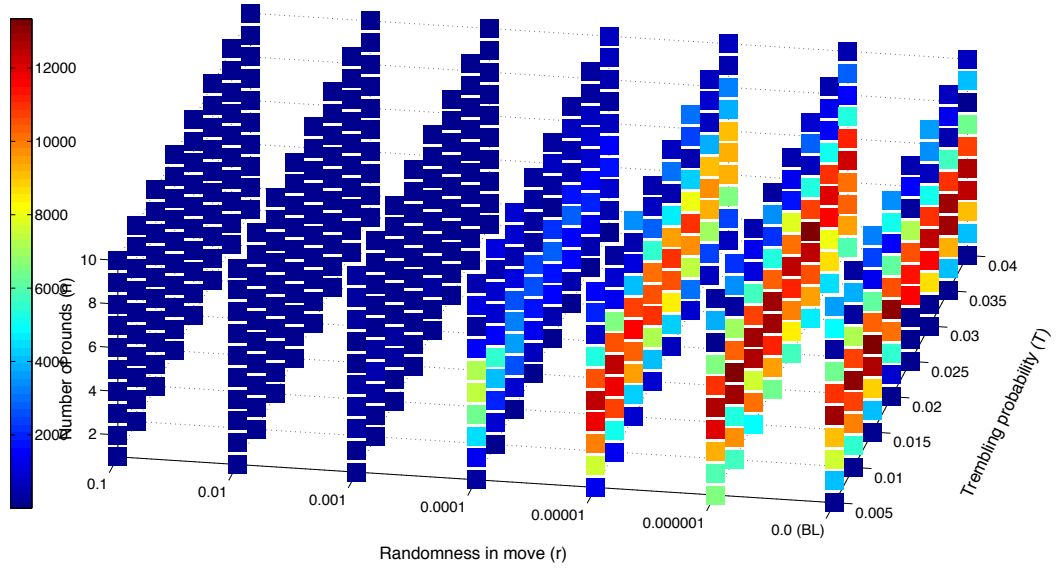


Figure A.77: *Error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n (implementation and perception errors).

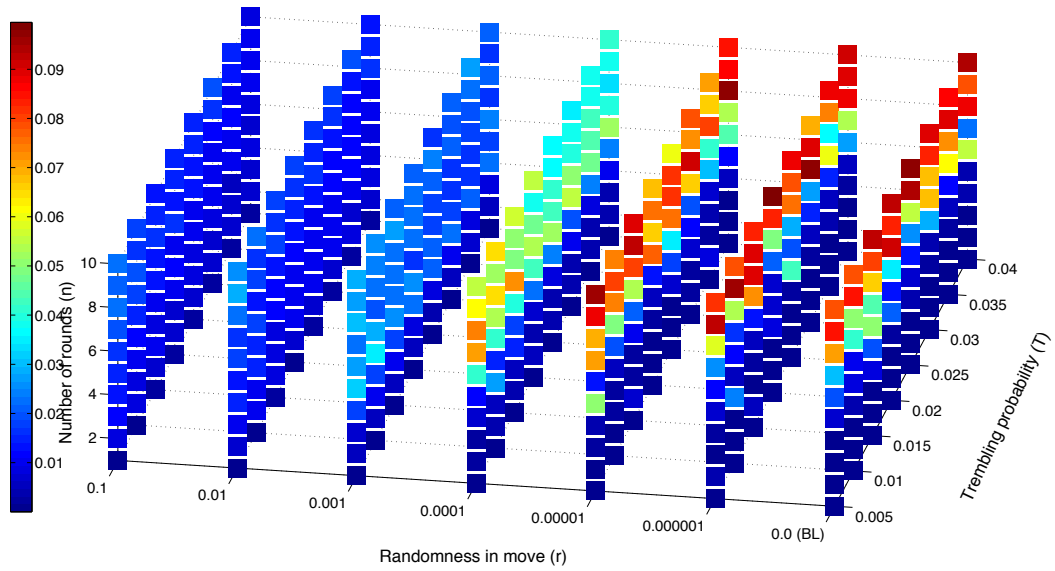


Figure A.78: *Error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (implementation and perception errors).

Error at 95% confidence for average payoff

Figures A.79 and A.80 show the *error at* 95% *confidence for average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Figures A.81 and A.82 show the *error at* 95% *confidence for average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Error at 95% confidence for average maximum payoff

Figures A.83 and A.84 show the *error at* 95% *confidence for average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Figures A.85 and A.86 show the *error at* 95% *confidence for average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

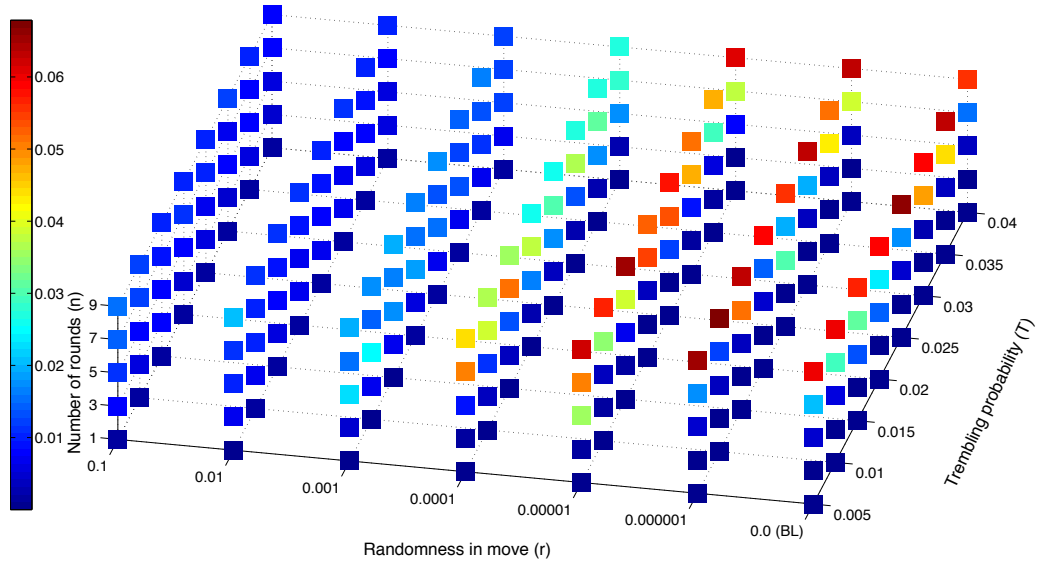


Figure A.79: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

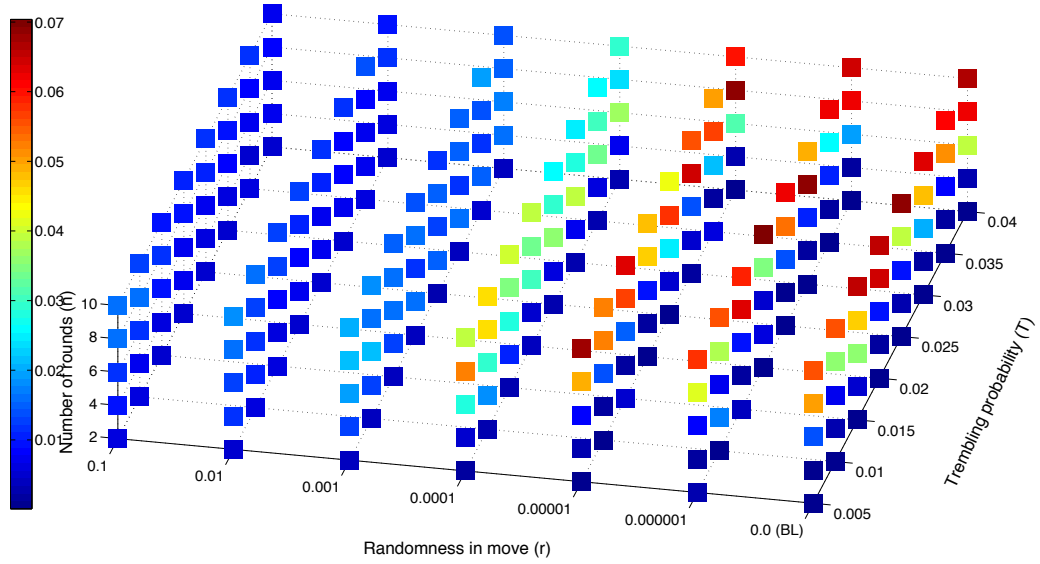


Figure A.80: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

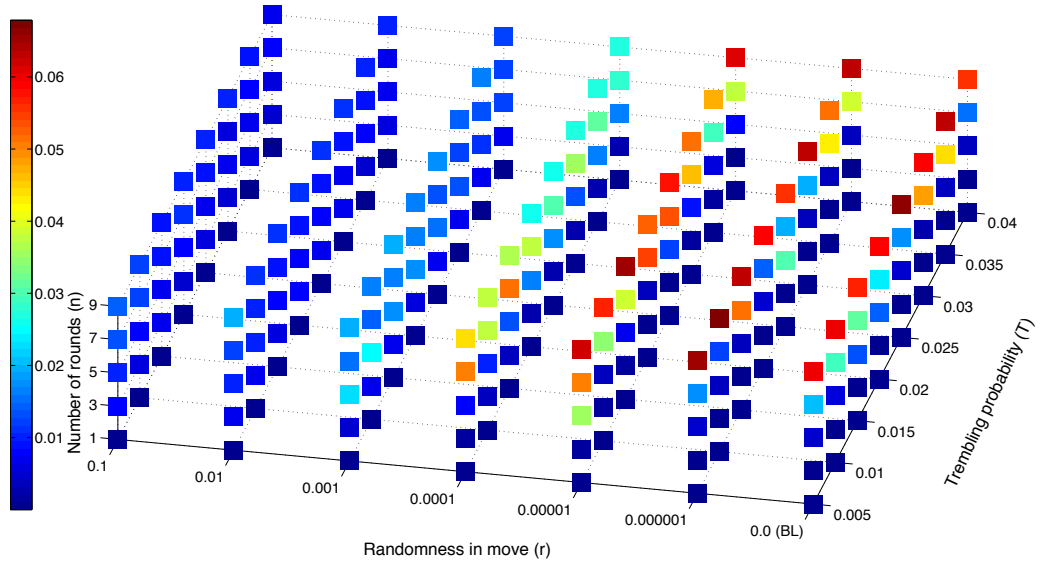


Figure A.81: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

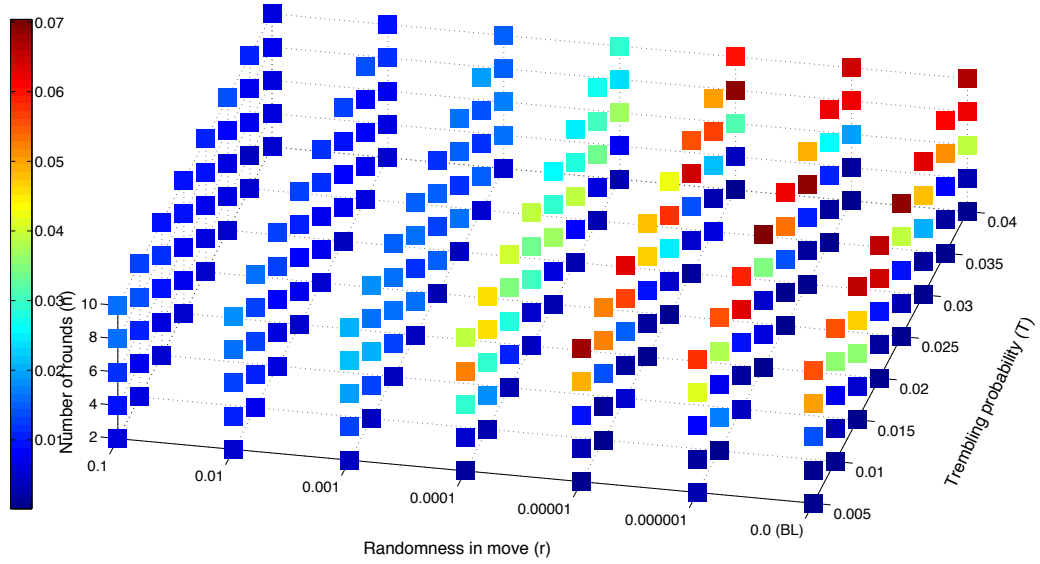


Figure A.82: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

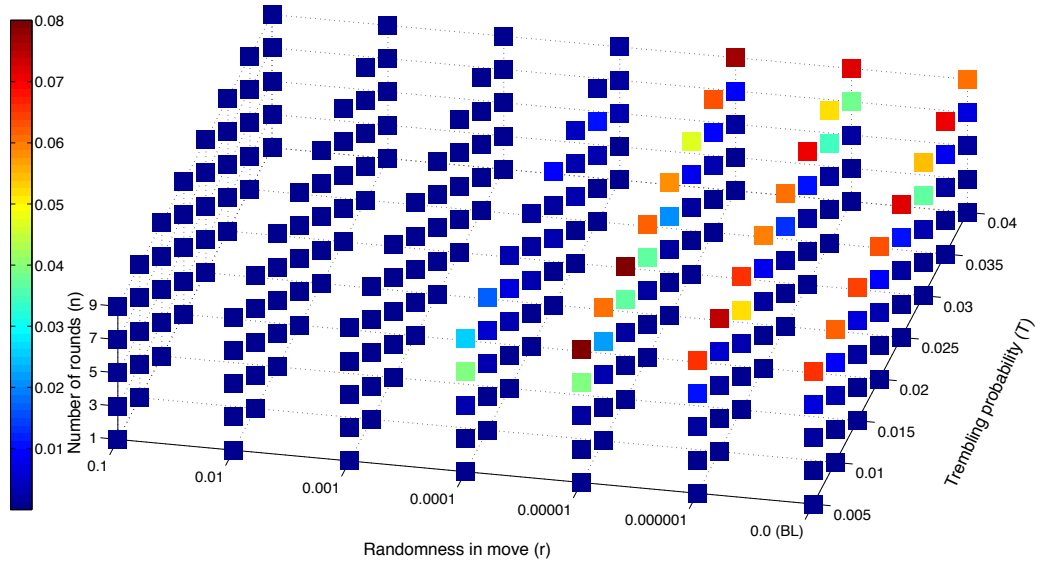


Figure A.83: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

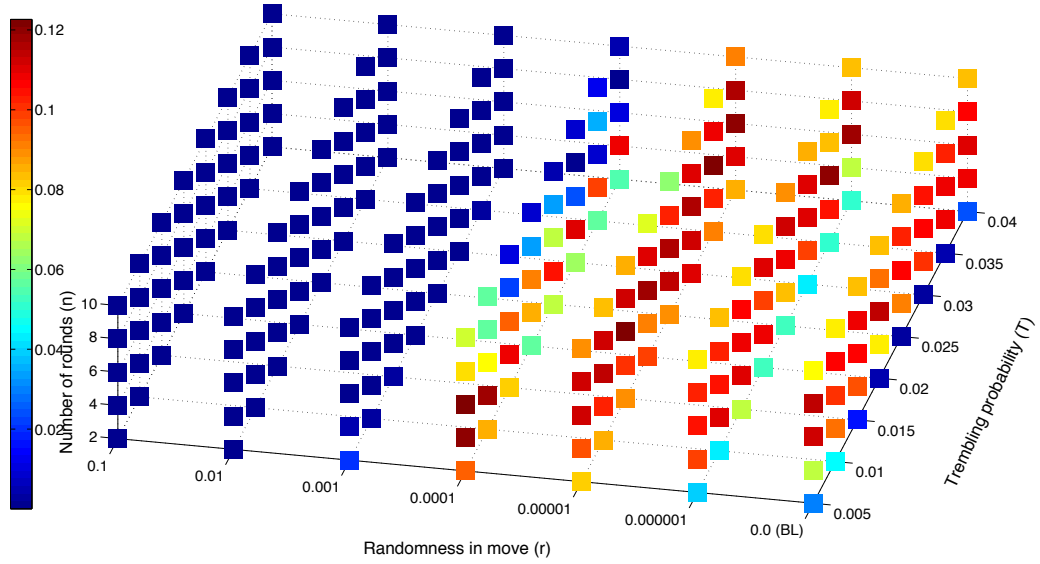


Figure A.84: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

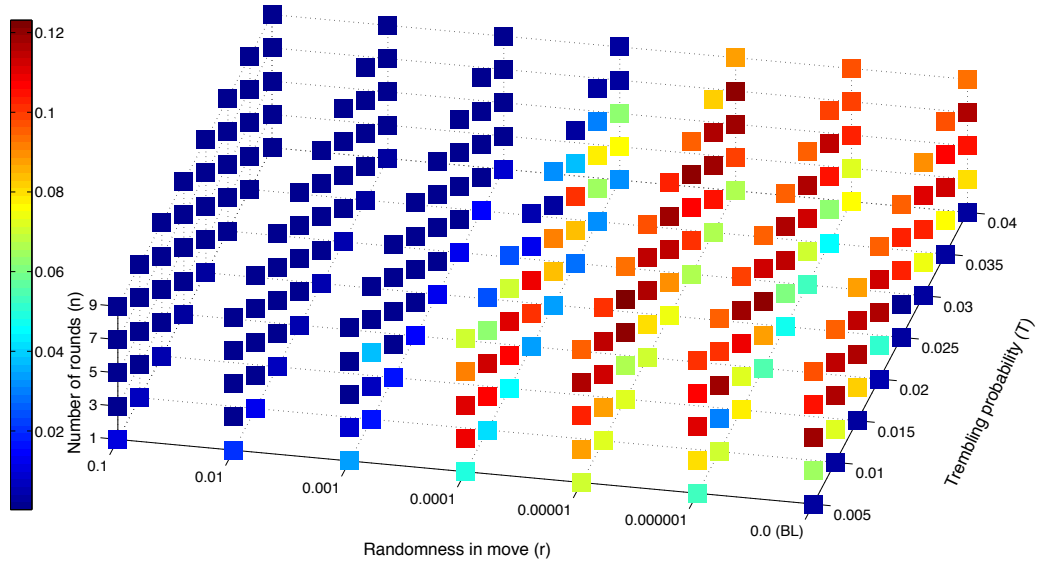


Figure A.85: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

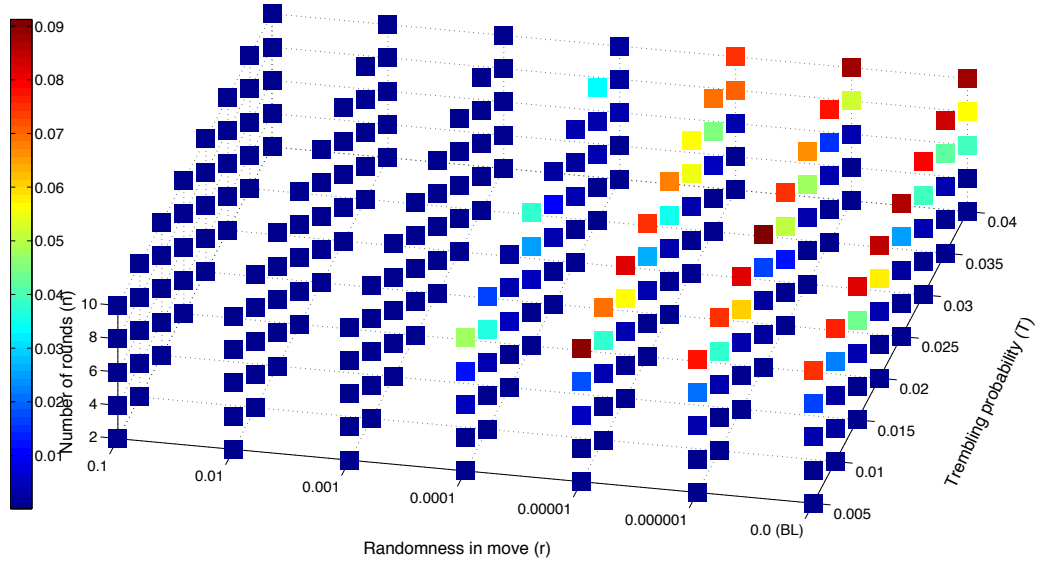


Figure A.86: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

Error at 95% confidence for average minimum payoff

Figures A.87 and A.88 show the *error at* 95% *confidence for average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

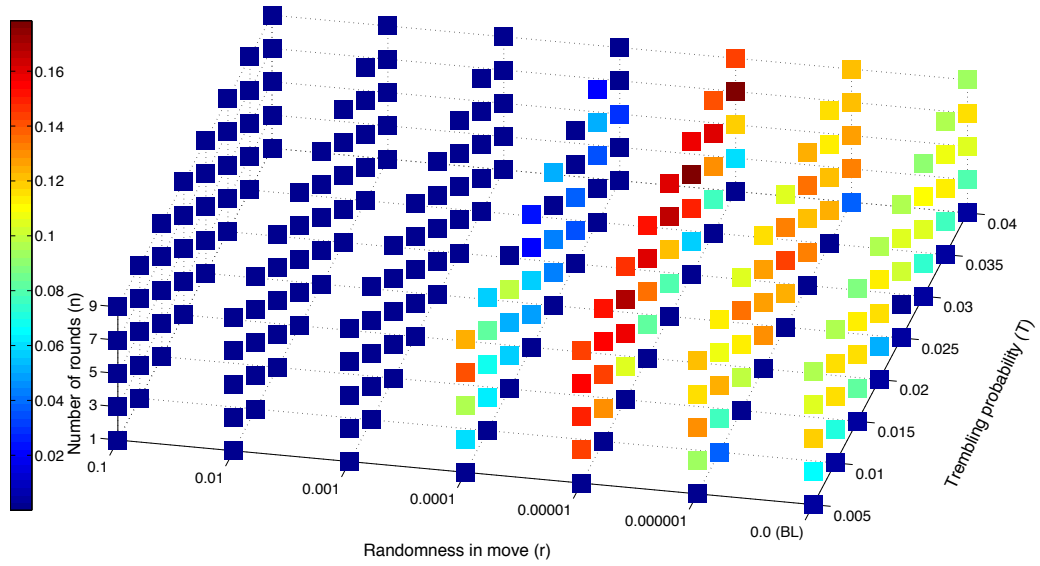


Figure A.87: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

Figures A.89 and A.90 show the *error at* 95% *confidence for average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

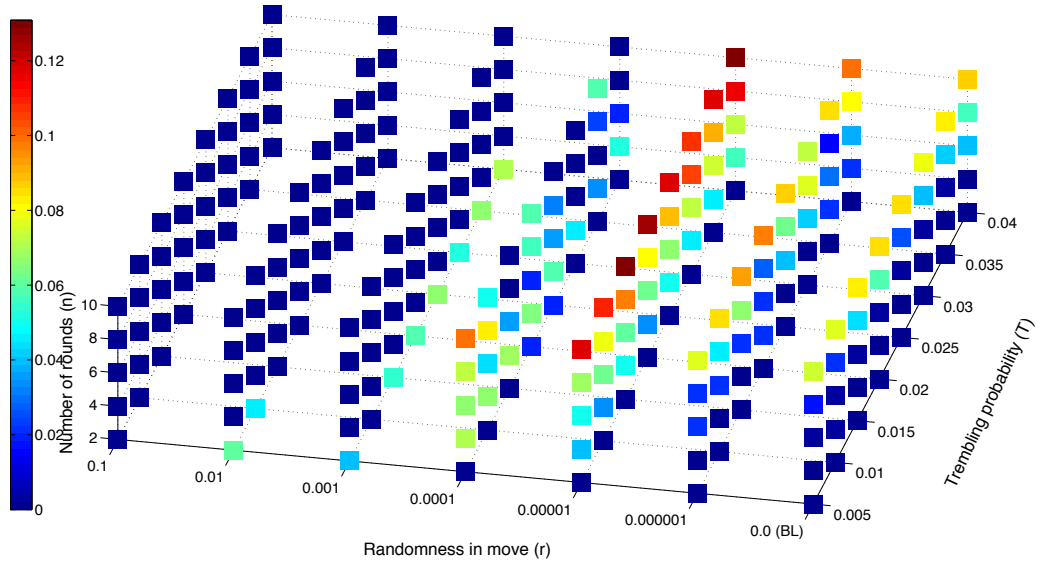


Figure A.88: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

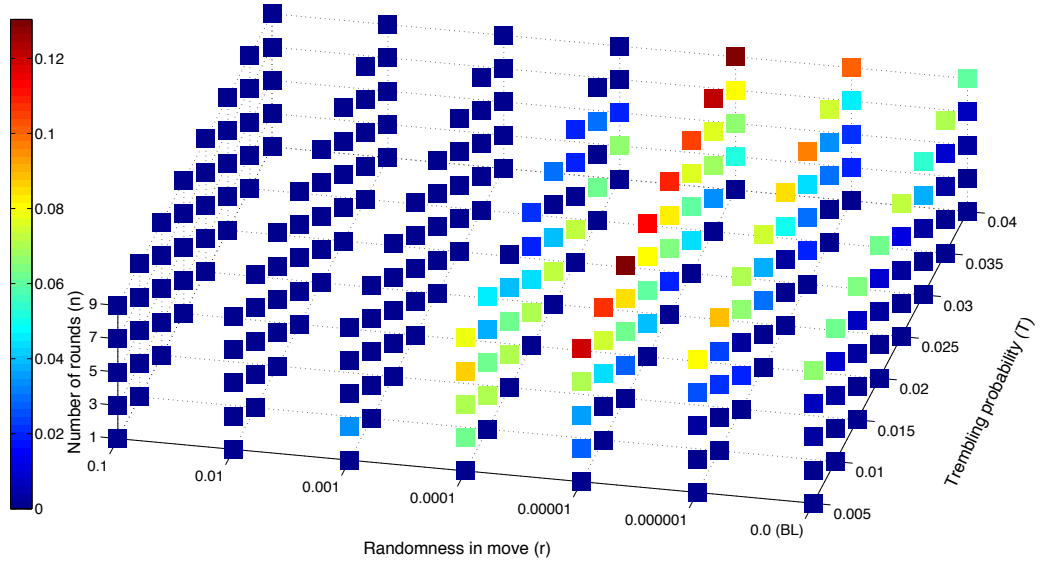


Figure A.89: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

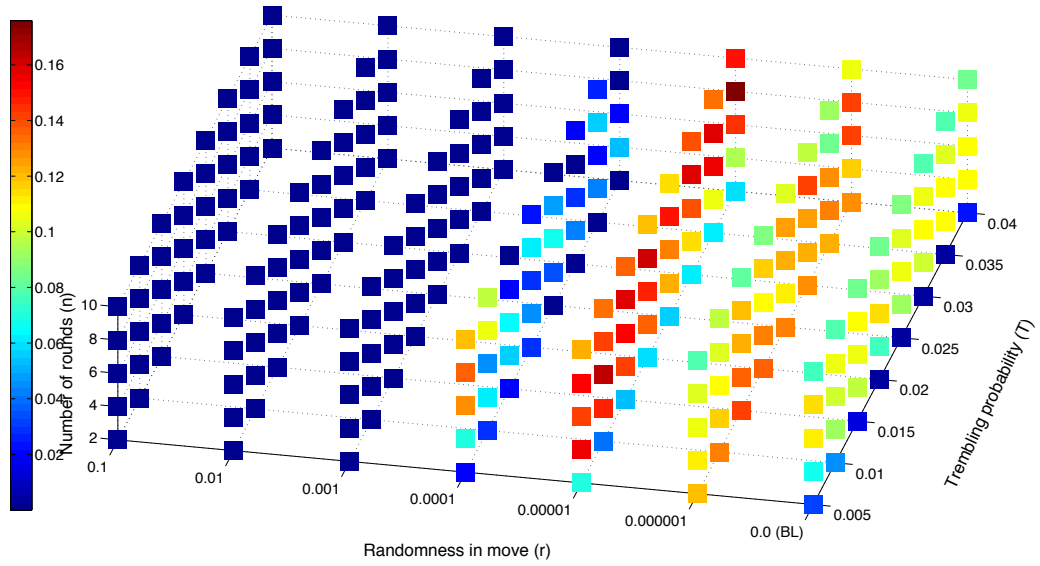


Figure A.90: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

Appendix B

Results for Randomness in Moves (r) Specified Using a Uniform Distribution in Chapter 4

We defined three types of rationality bounds in Chapter 4, viz. implementation errors, perception errors, and a combination of the two. This Appendix shows the results obtained when the r (randomness in moves) parameter that partly specifies the bounds in terms of Gaussian noise, now specifies the bounds in terms of Uniformly distributed noise. This results in the action or perception of an agent to have Uniformly distributed noise within the interval $[-r\sqrt{3}, r\sqrt{3}]$ added to it.

We show three types of graphs in this Appendix. The first type, in Section B.1, show the general trends of the evaluation metrics that we considered in Chapter 4, with respect to variations in r, for the three types of rationality bounds. The second type, in Section B.2, show the values of the evaluation metric, together with errors at a 95% confidence level, with respect to variations in r, T and n. These graphs are of the same form as those plotted for the Gaussian noise case in Appendix A (Section A.2), but for Uniformly distributed noise. The third type, in Section B.3, show the errors at a 95% confidence level separately as well, again with respect to variations in r, T and n. These graphs are of the same form as those plotted for the Gaussian noise case in Appendix A (Section A.3), but for Uniformly distributed noise. Please read Chapter 4 (Section 4.4.3) to learn how to refer to the graphs in Section B.1, and Appendix A to learn how to refer to

graphs in Sections B.2 and B.3.

B.1 Averages Showing Trends

B.1.1 Implementation Errors

Average first hitting time

Figure B.1 shows the *average first hitting time*, across 50 runs, averaged further across T, for all values of r and n, for implementation errors.

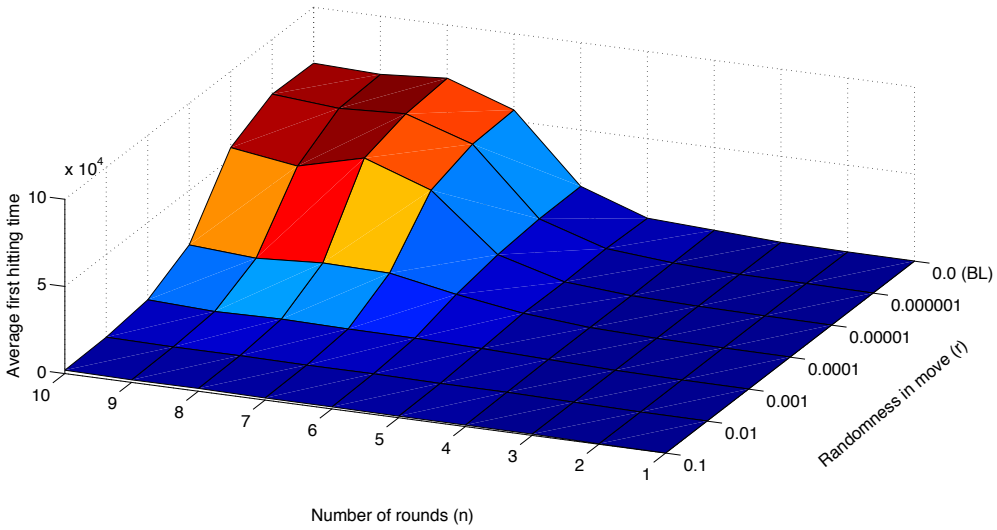


Figure B.1: Average first hitting time, across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average stay in equilibrium

Figure B.2 shows the *average stay within ϵ distance from equilibrium*, across 50 runs, averaged further across T, for all values of r and n, for implementation errors.

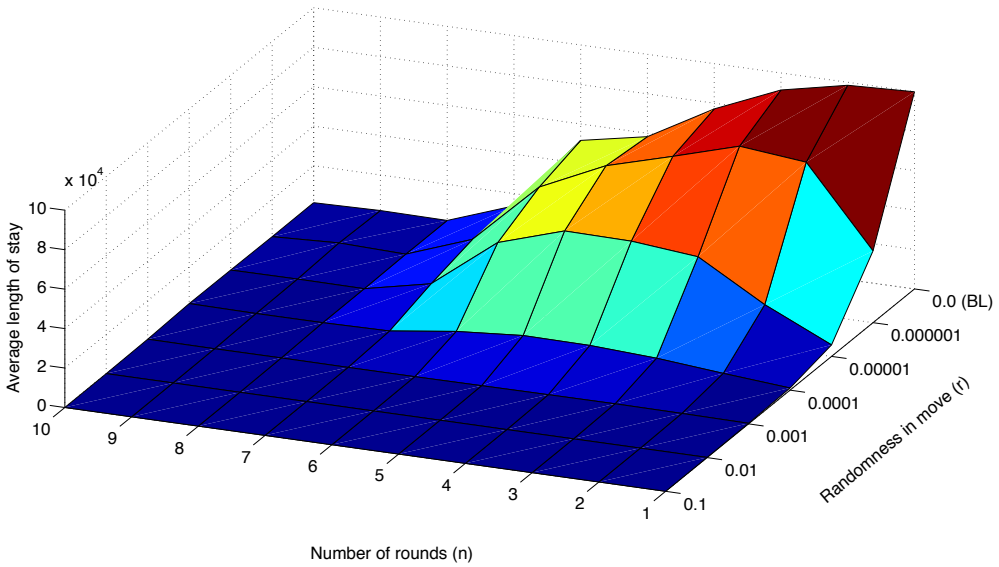


Figure B.2: Average length of stay in equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average distance from equilibrium

Figure B.3 shows the *average distance from equilibrium, after first hit*, across 50 runs, averaged further across T, for all values of r and n, for implementation errors.

Average payoff

Figure B.4 shows the *average payoff for Player 1, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation errors.

Figure B.5 shows the *average payoff for Player 2, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation errors.

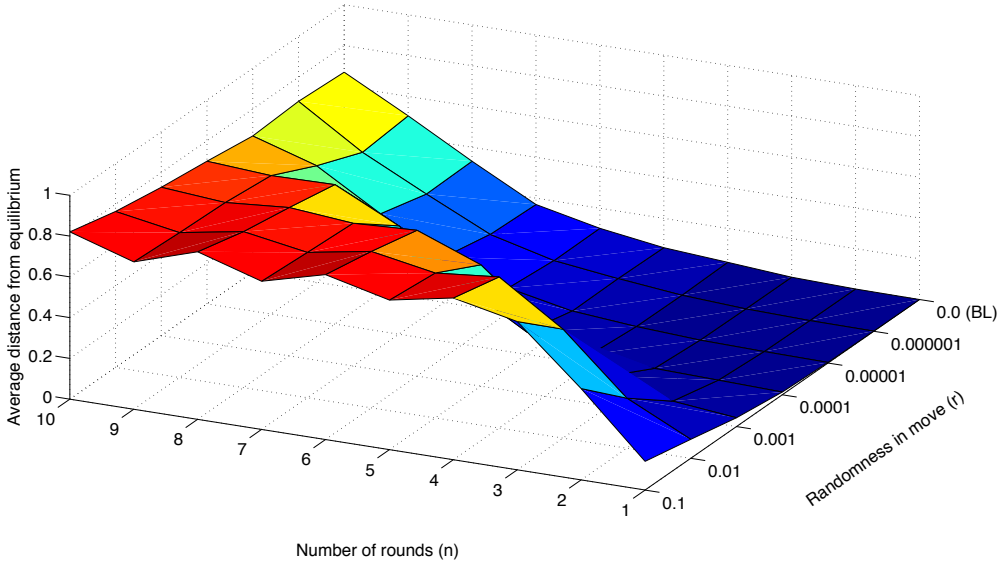


Figure B.3: Average distance from the equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average maximum payoff

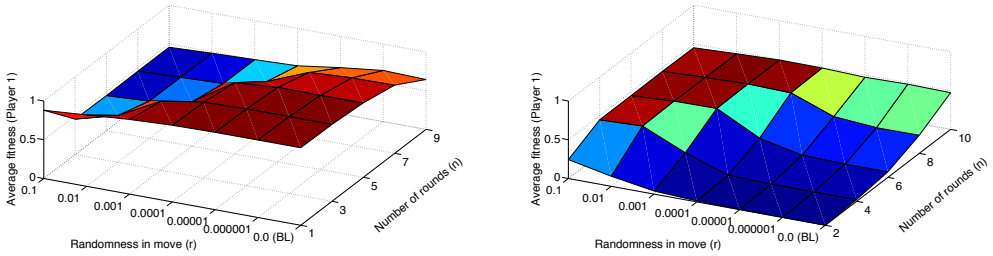
Figure B.6 shows the *average maximum payoff for Player 1, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation errors.

Figure B.7 shows the *average maximum payoff for Player 2, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation errors.

Average minimum payoff

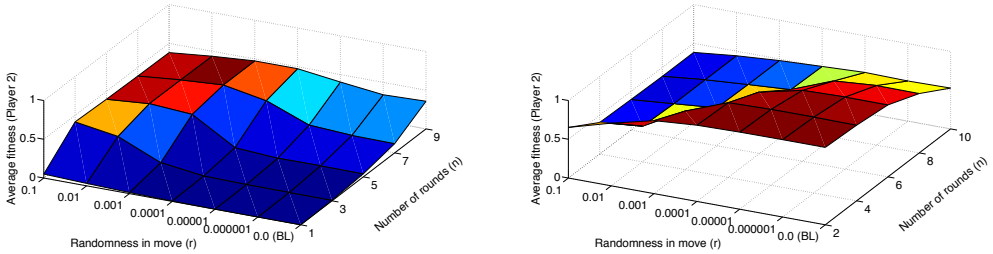
Figure B.8 shows the *average minimum payoff for Player 1, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation errors.

Figure B.9 shows the *average minimum payoff for Player 2, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation errors.



(a) n odd. (b) n even.

Figure B.4: Average payoff of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.



(a) n odd. (b) n even.

Figure B.5: Average payoff of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

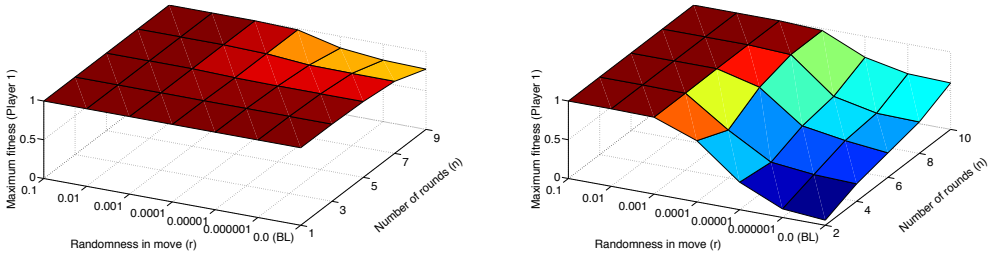
B.1.2 Perception Errors

Average first hitting time

Figure B.10 shows the *average first hitting time*, across 50 runs, averaged further across T, for all values of r and n, for perception errors.

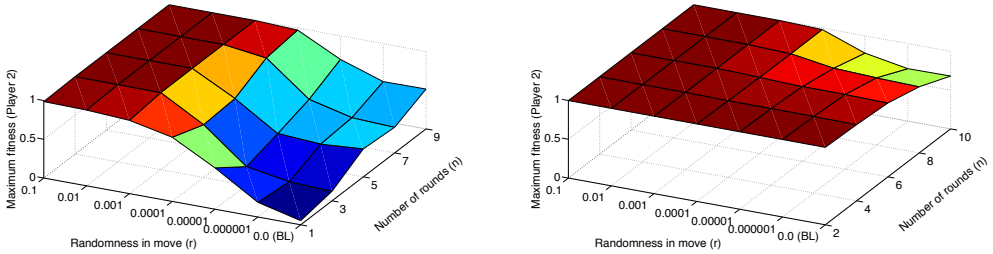
Average stay in equilibrium

Figure B.11 shows the *average stay within ϵ distance from equilibrium*, across 50 runs, averaged further across T, for all values of r and n, for perception errors.



(a) n odd. (b) n even.

Figure B.6: Average maximum payoff of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.



(a) n odd. (b) n even.

Figure B.7: Average maximum payoff of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

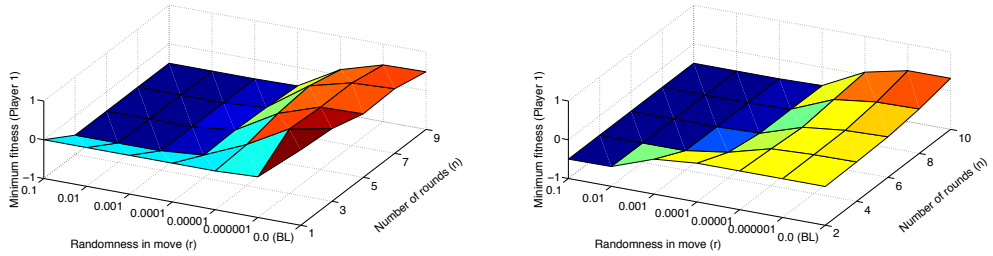
Average distance from equilibrium

Figure B.12 shows the *average distance from equilibrium, after first hit*, across 50 runs, averaged further across T, for all values of r and n, for perception errors.

Average payoff

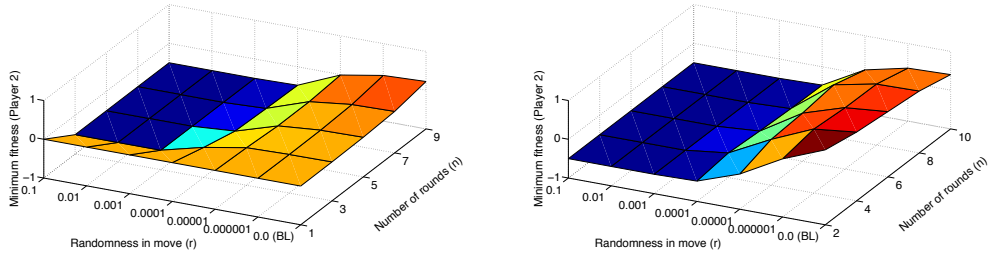
Figure B.13 shows the *average payoff for Player 1, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for perception errors.

Figure B.14 shows the *average payoff for Player 2, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even,



(a) n odd. (b) n even.

Figure B.8: Average minimum payoff of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.



(a) n odd. (b) n even.

Figure B.9: Average minimum payoff of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

for perception errors.

Average maximum payoff

Figure B.15 shows the *average maximum payoff for Player 1, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for perception errors.

Figure B.16 shows the *average maximum payoff for Player 2, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for perception errors.

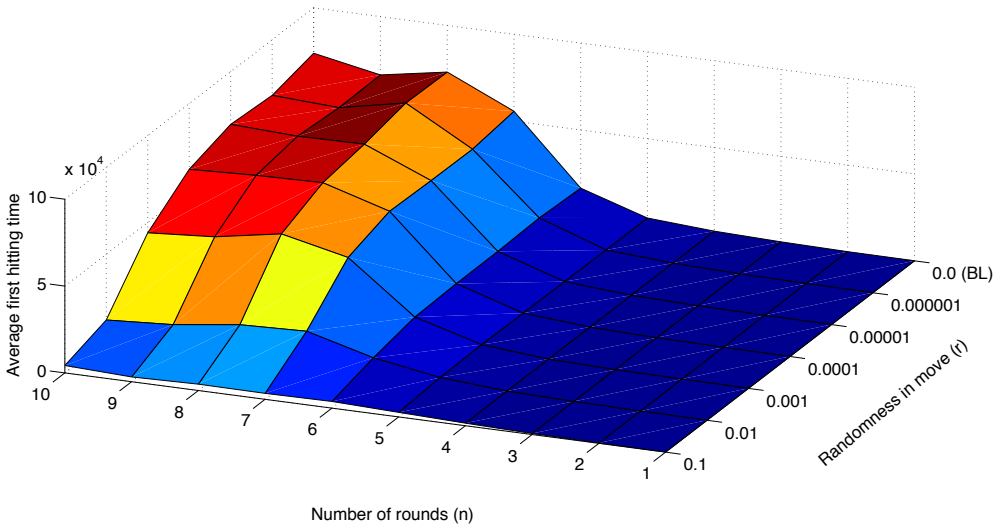


Figure B.10: Average first hitting time, across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

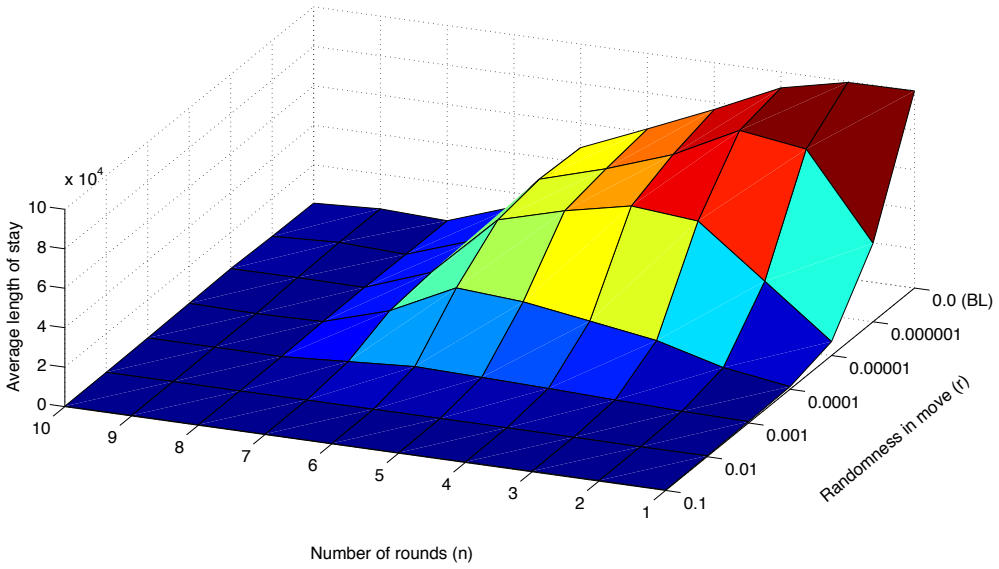


Figure B.11: Average length of stay in equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

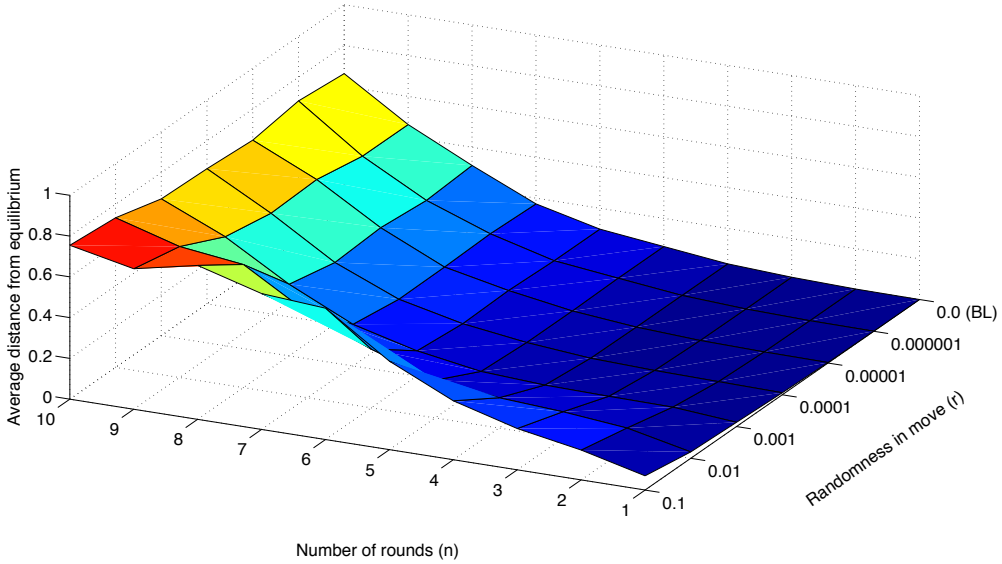


Figure B.12: Average distance from the equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average minimum payoff

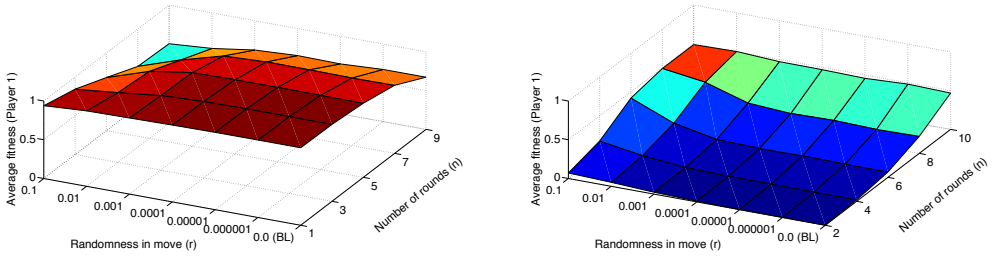
Figure B.17 shows the *average minimum payoff for Player 1, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for perception errors.

Figure B.18 shows the *average minimum payoff for Player 2, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for perception errors.

B.1.3 Implementation and Perception Errors

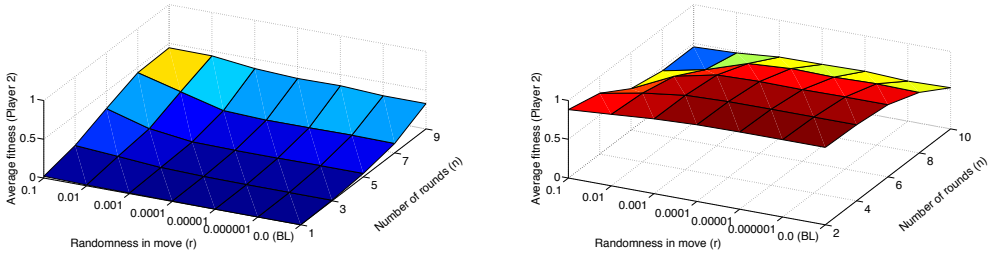
Average first hitting time

Figure B.19 shows the *average first hitting time*, across 50 runs, averaged further across T, for all values of r and n, for implementation and perception errors.



(a) n odd. (b) n even.

Figure B.13: Average payoff of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.



(a) n odd. (b) n even.

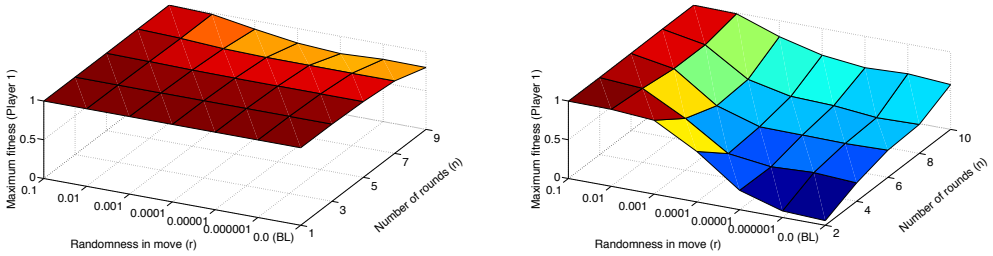
Figure B.14: Average payoff of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average stay in equilibrium

Figure B.20 shows the *average stay within ϵ distance from equilibrium*, across 50 runs, averaged further across T, for all values of r and n, for implementation and perception errors.

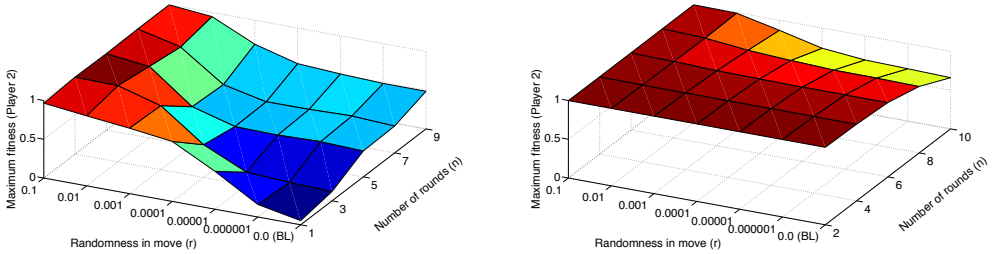
Average distance from equilibrium

Figure B.21 shows the *average distance from equilibrium, after first hit*, across 50 runs, averaged further across T, for all values of r and n, for implementation and perception errors.



(a) n odd. (b) n even.

Figure B.15: Average maximum payoff of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.



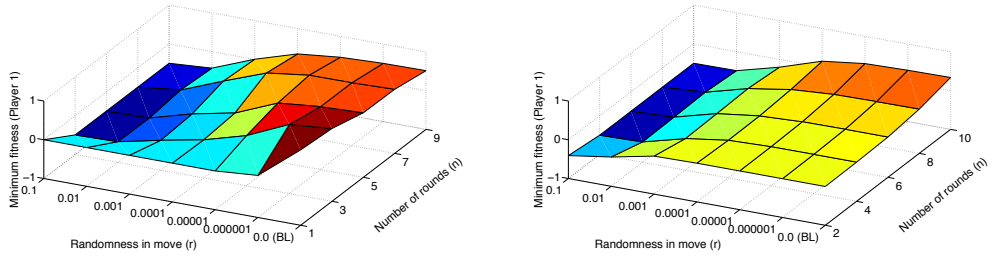
(a) n odd. (b) n even.

Figure B.16: Average maximum payoff of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average payoff

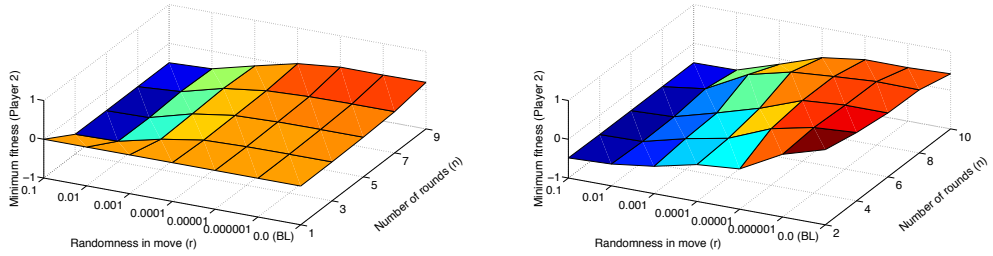
Figure B.22 shows the *average payoff for Player 1, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation and perception errors.

Figure B.23 shows the *average payoff for Player 2, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation and perception errors.



(a) n odd. (b) n even.

Figure B.17: Average minimum payoff of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.



(a) n odd. (b) n even.

Figure B.18: Average minimum payoff of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average maximum payoff

Figure B.24 shows the *average maximum payoff for Player 1, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation and perception errors.

Figure B.25 shows the *average maximum payoff for Player 2, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation and perception errors.

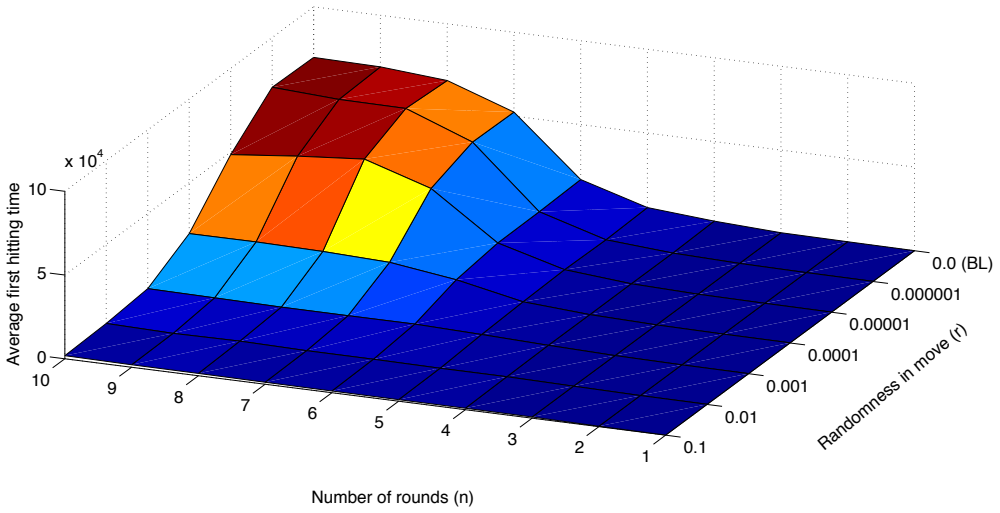


Figure B.19: Average first hitting time, across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

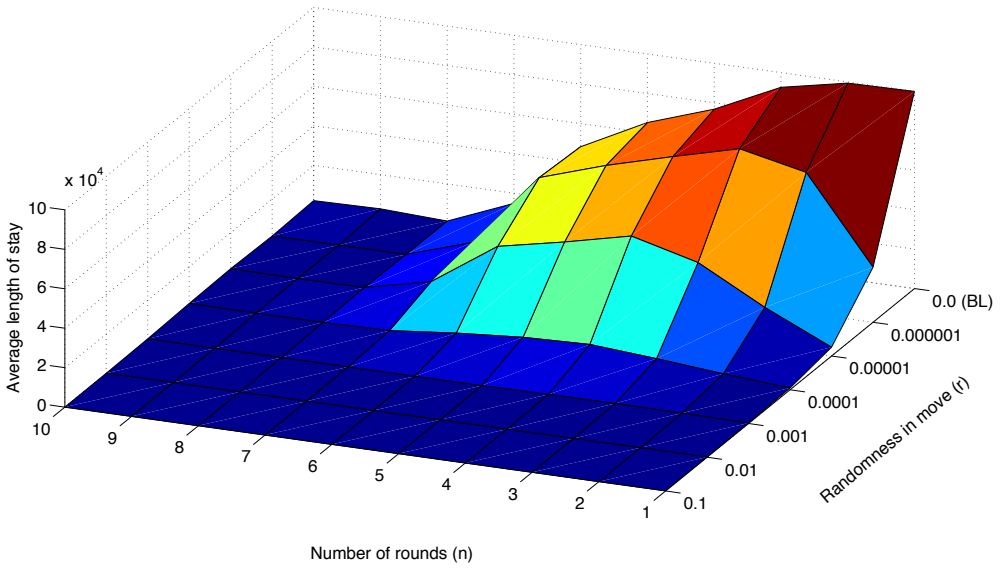


Figure B.20: Average length of stay in equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

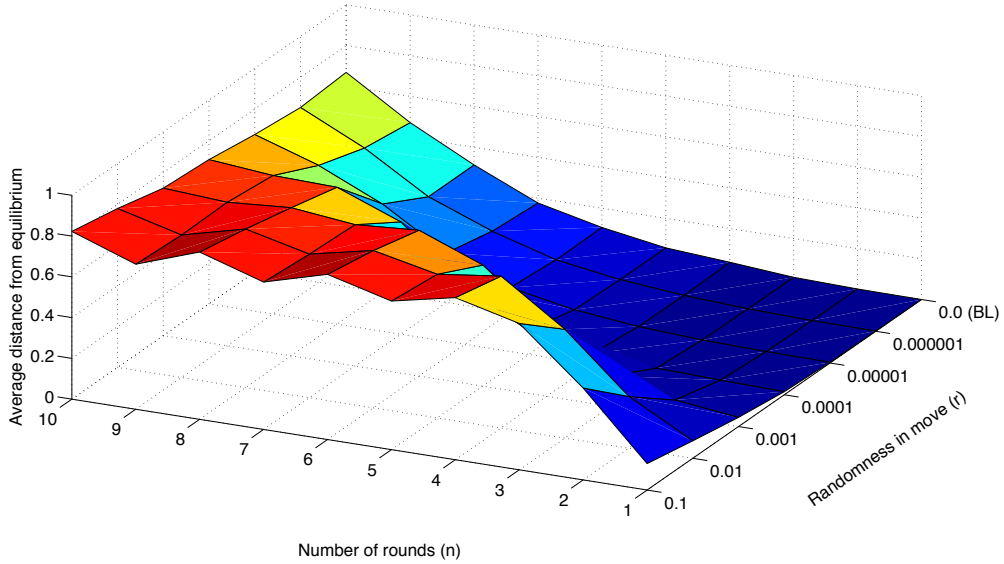


Figure B.21: Average distance from the equilibrium after first hit, across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average minimum payoff

Figure B.26 shows the *average minimum payoff for Player 1, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation and perception errors.

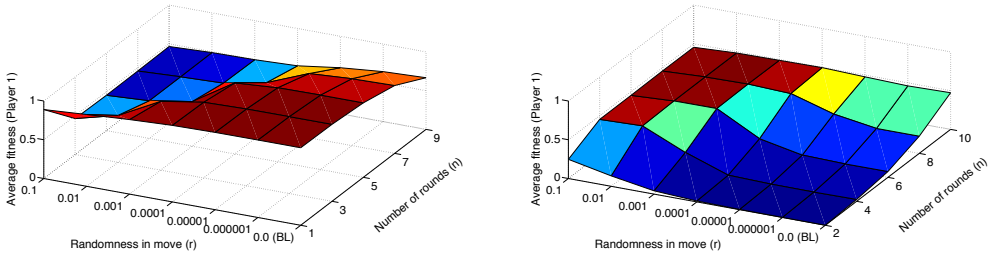
Figure B.27 shows the *average minimum payoff for Player 2, after first hit*, across 50 runs, averaged further across T, for all values of r, and for n being odd and even, for implementation and perception errors.

B.2 Averages with Confidence

B.2.1 Implementation Errors

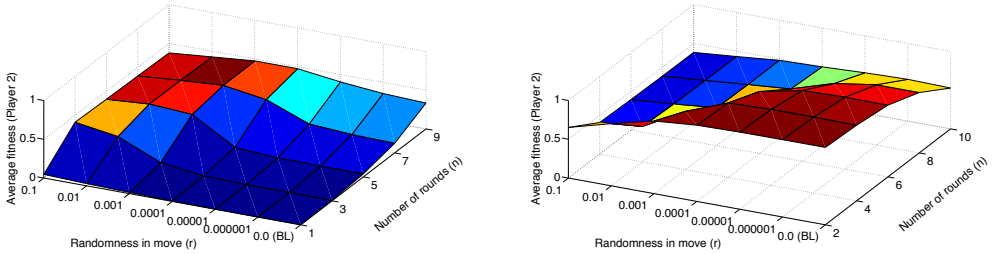
Average first hitting time

Figure B.28 shows the *average first hitting time*, across 50 runs for all values of r, T and n, for implementation errors.



(a) n odd. (b) n even.

Figure B.22: Average payoff of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.



(a) n odd. (b) n even.

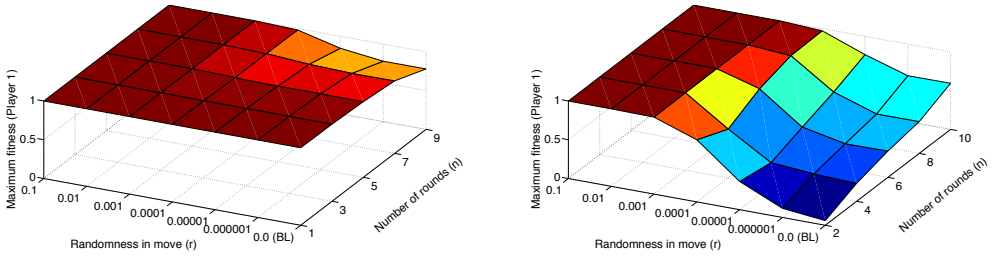
Figure B.23: Average payoff of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average stay in equilibrium

Figure B.29 shows the *average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n, for implementation errors.

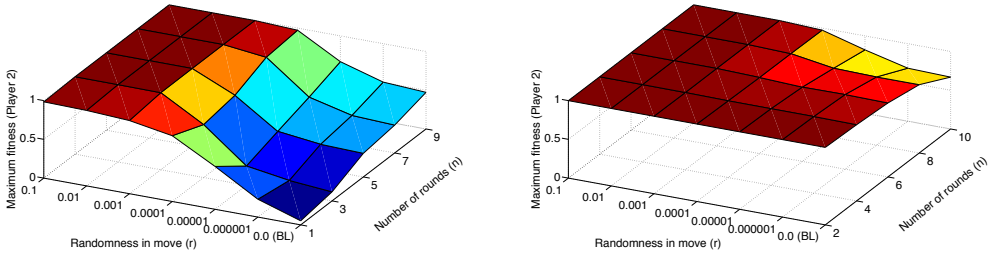
Average distance from equilibrium

Figure B.30 shows the *average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for implementation errors.



(a) n odd. (b) n even.

Figure B.24: Average maximum payoff of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.



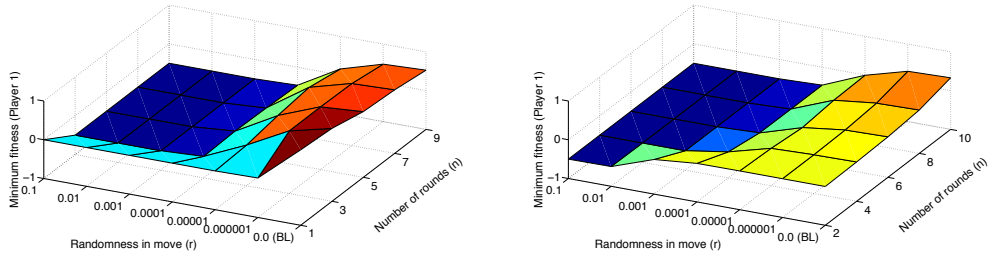
(a) n odd. (b) n even.

Figure B.25: Average maximum payoff of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average payoff

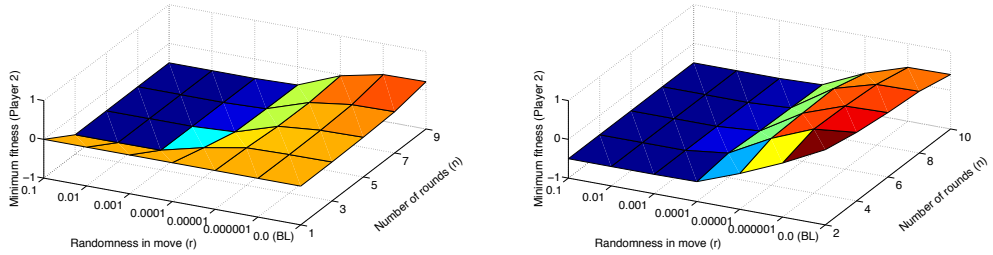
Figures B.31 and B.32 show the *average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Figures B.33 and B.34 show the *average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.



(a) n odd. (b) n even.

Figure B.26: Average minimum payoff of Player 1 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.



(a) n odd. (b) n even.

Figure B.27: Average minimum payoff of Player 2 across generations since first hit, averaged across 50 runs for all values of r and n, averaged further across T (implementation and perception errors). Change minimal across T as can be seen in the graphs in Sections B.2 and B.3.

Average maximum payoff

Figures B.35 and B.36 show the *average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Figures B.37 and B.38 show the *average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

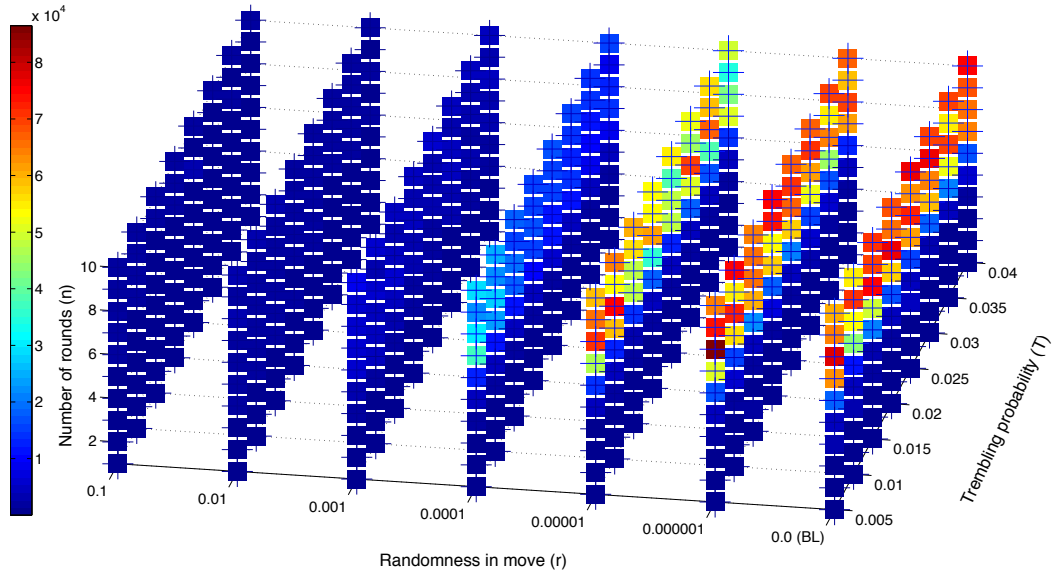


Figure B.28: *Average first hitting time*, across 50 runs for all values of r, T and n (implementation errors). See Figure B.73 for the magnitudes of errors in detail.

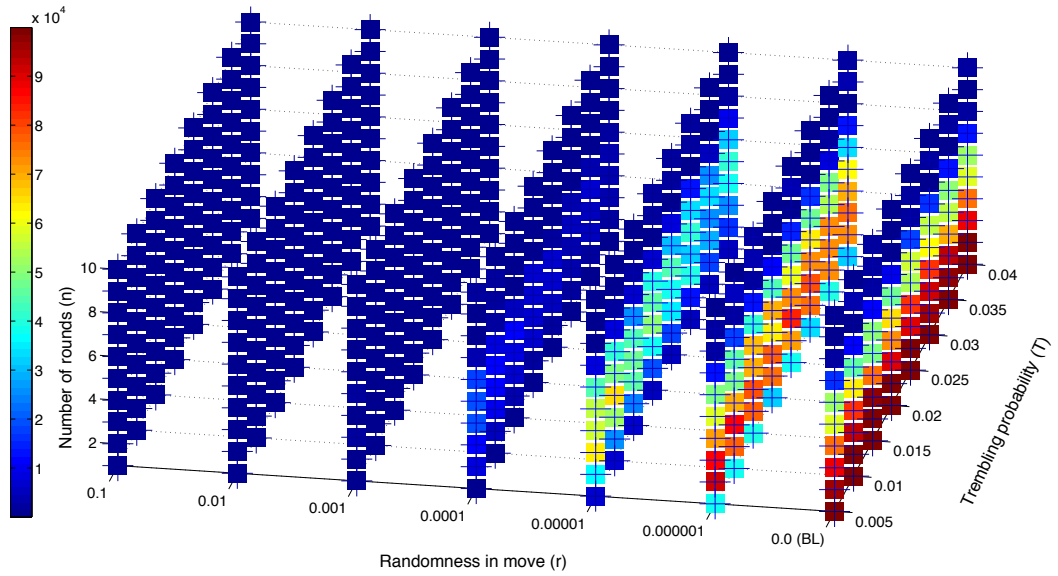


Figure B.29: *Average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n (implementation errors). See Figure B.74 for the magnitudes of errors in detail.

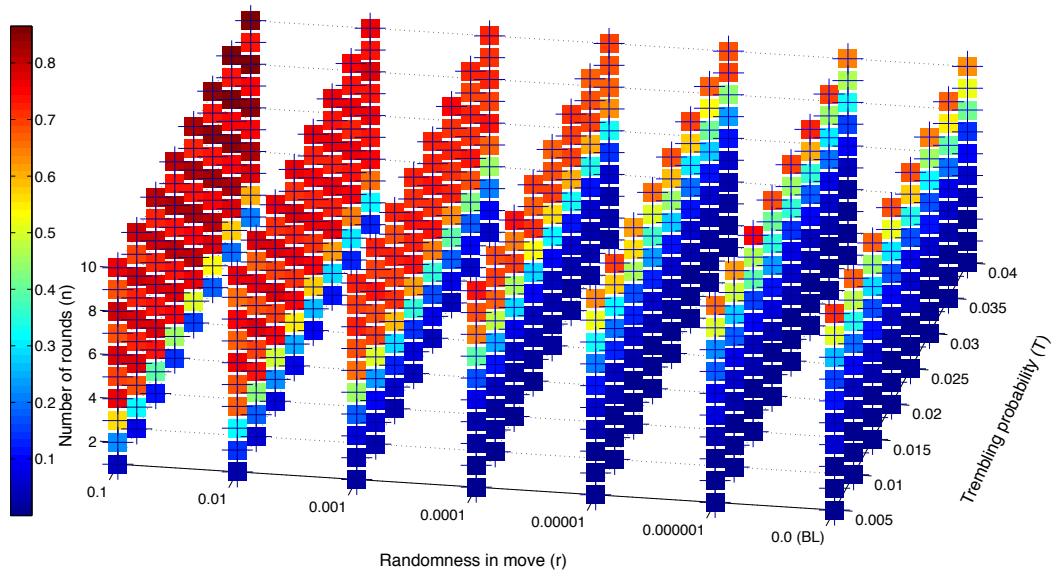


Figure B.30: *Average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (implementation errors). See Figure B.75 for the magnitudes of errors in detail.

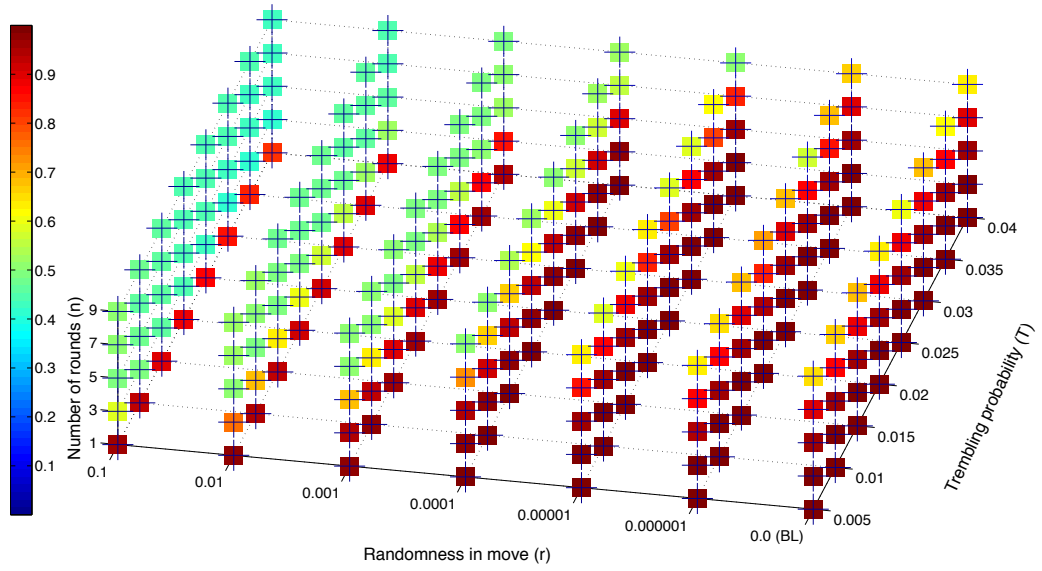


Figure B.31: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure B.76 for the magnitudes of errors in detail.

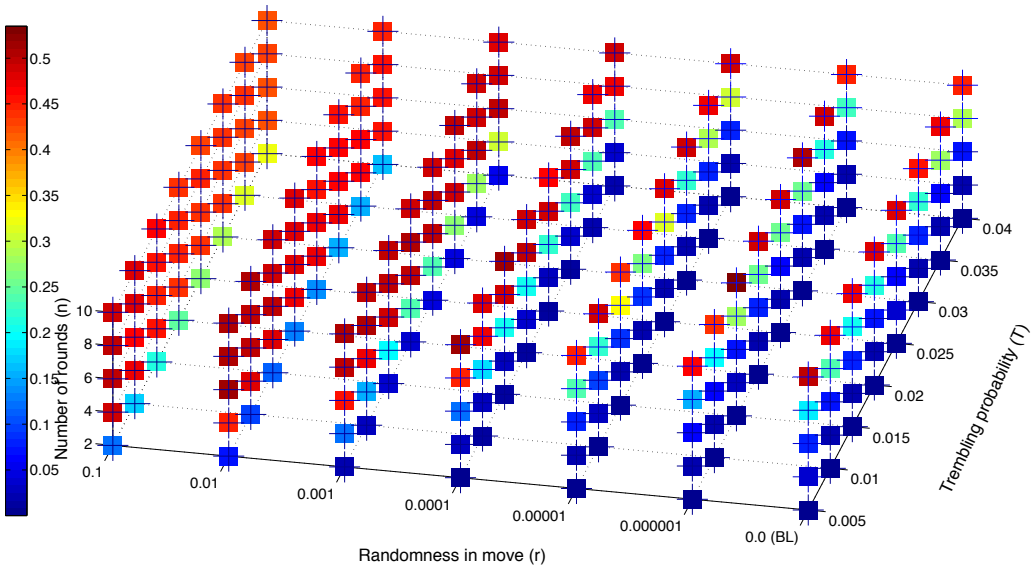


Figure B.32: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure B.77 for the magnitudes of errors in detail.

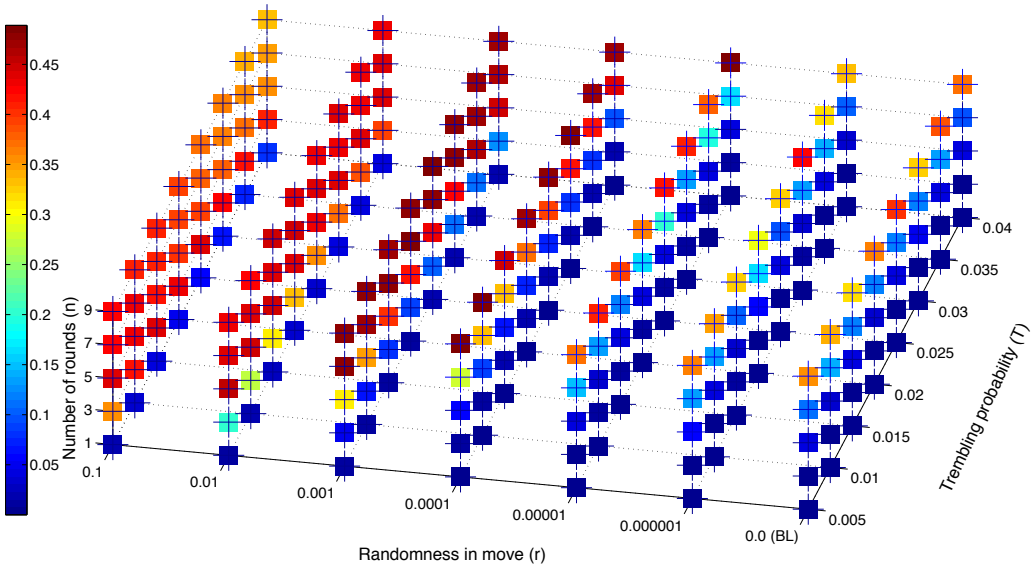


Figure B.33: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure B.78 for the magnitudes of errors in detail.

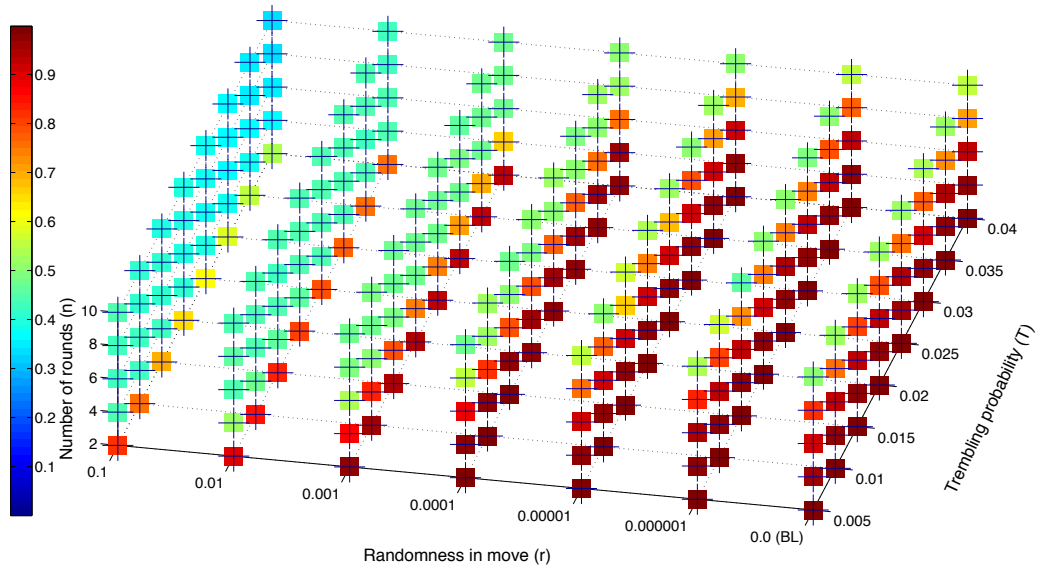


Figure B.34: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure B.79 for the magnitudes of errors in detail.

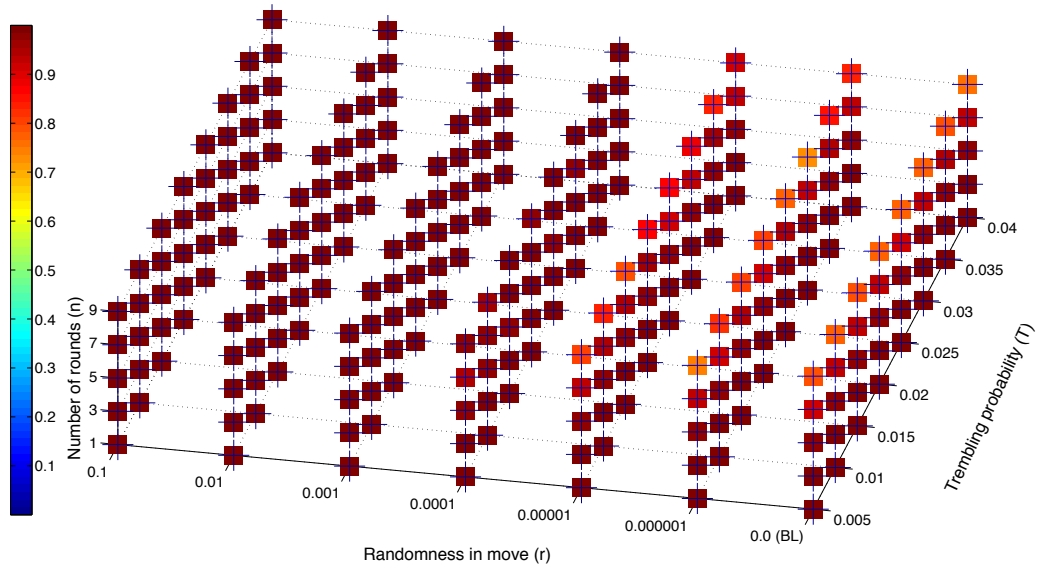


Figure B.35: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure B.80 for the magnitudes of errors in detail.

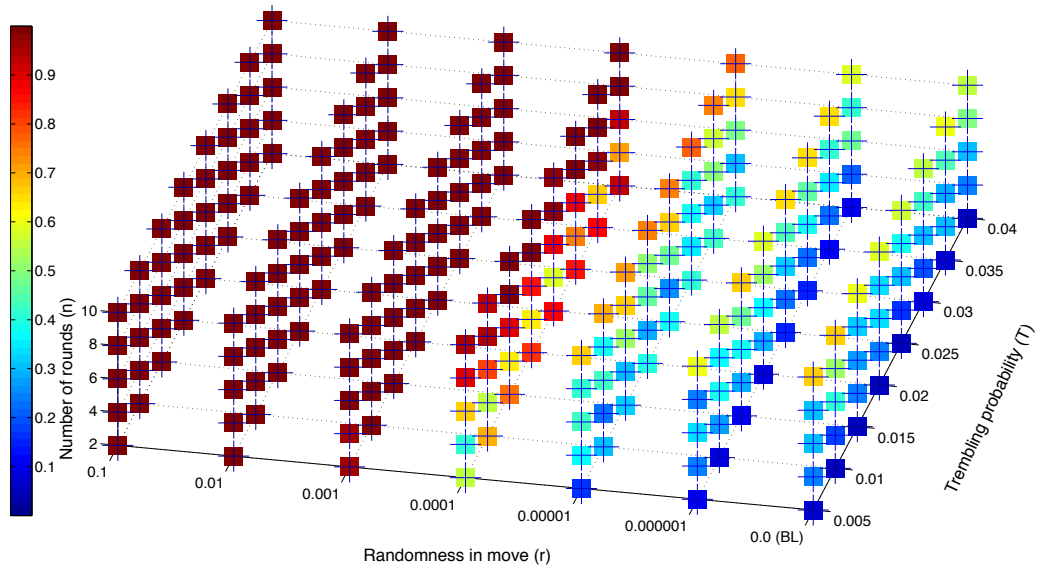


Figure B.36: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure B.81 for the magnitudes of errors in detail.

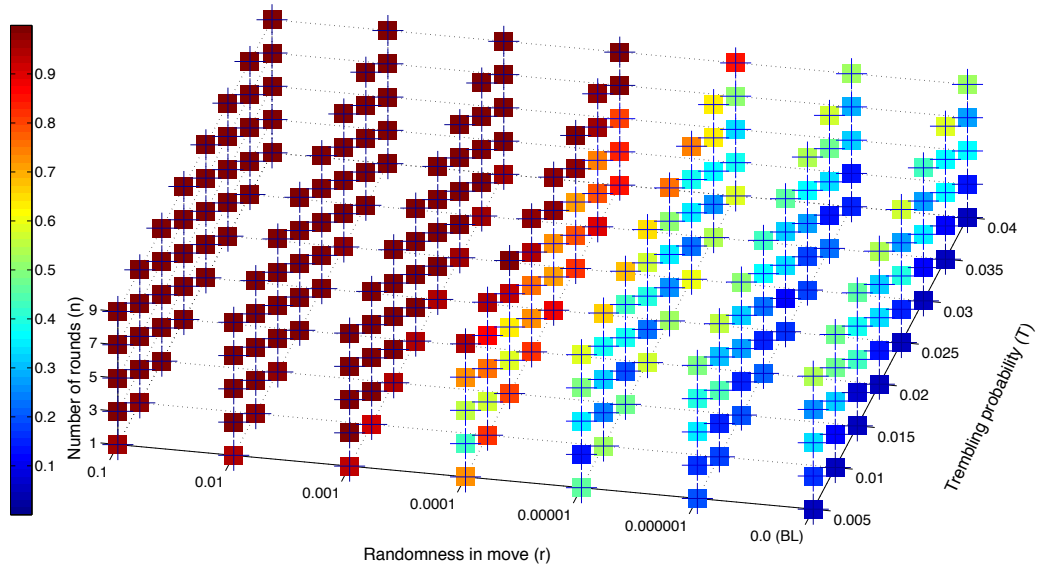


Figure B.37: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure B.82 for the magnitudes of errors in detail.

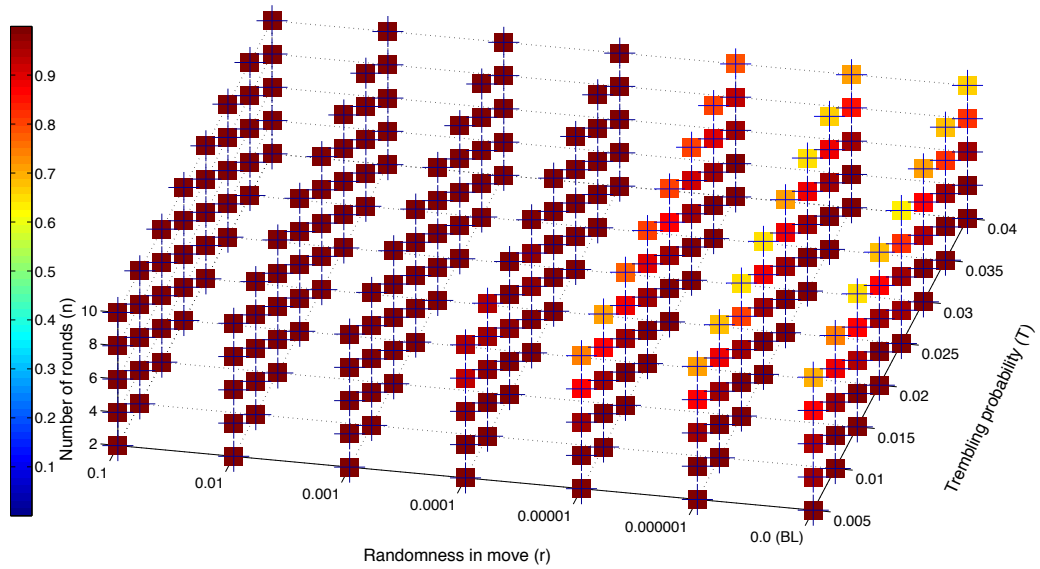


Figure B.38: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure B.83 for the magnitudes of errors in detail.

Average minimum payoff

Figures B.39 and B.40 show the *average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Figures B.41 and B.42 show the *average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

B.2.2 Perception Errors

Average first hitting time

Figure B.43 shows the *average first hitting time*, across 50 runs for all values of r, T and n, for perception errors.

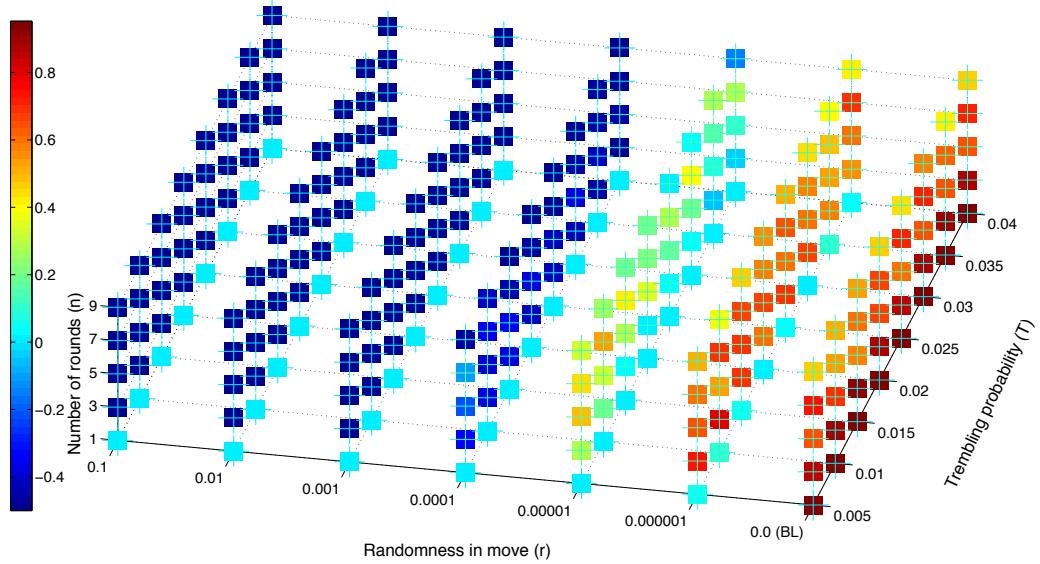


Figure B.39: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure B.84 for the magnitudes of errors in detail.

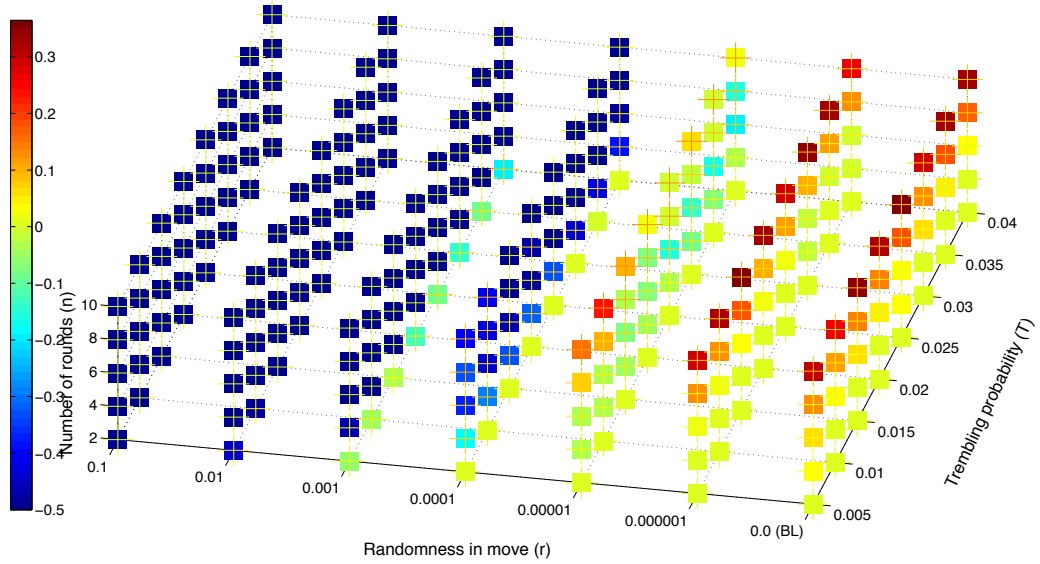


Figure B.40: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure B.85 for the magnitudes of errors in detail.

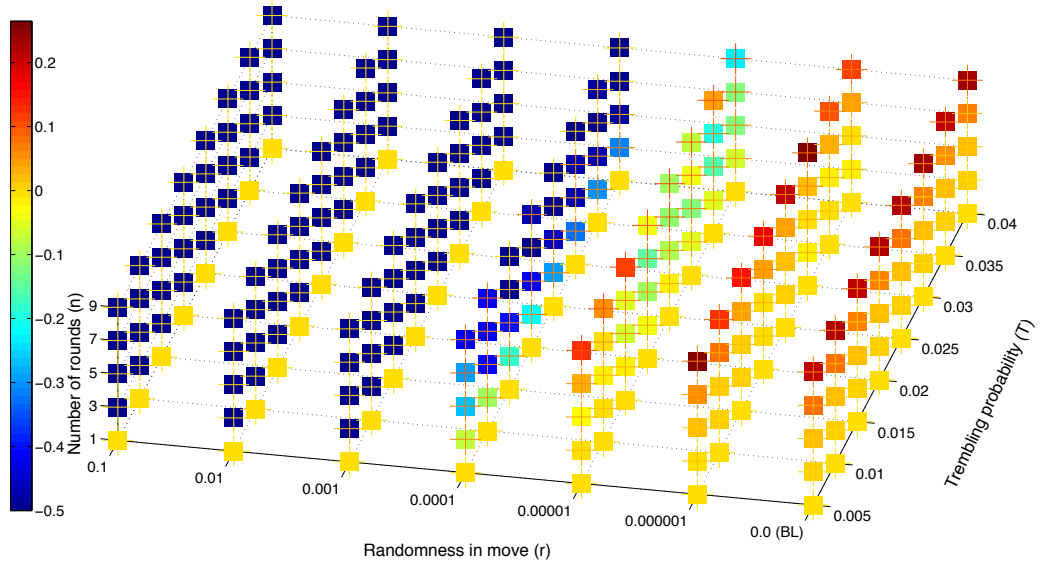


Figure B.41: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors). See Figure B.86 for the magnitudes of errors in detail.

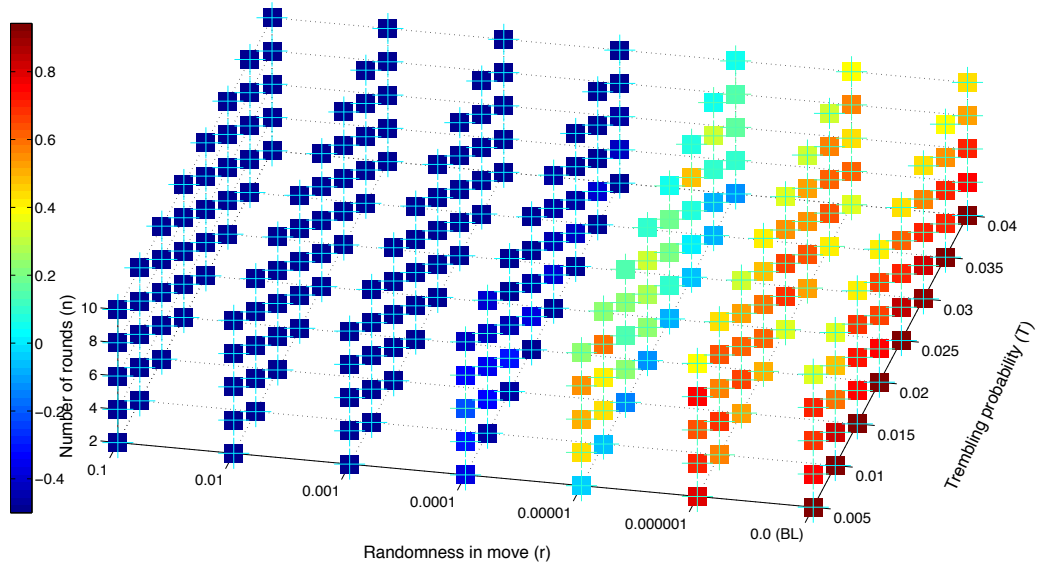


Figure B.42: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors). See Figure B.87 for the magnitudes of errors in detail.

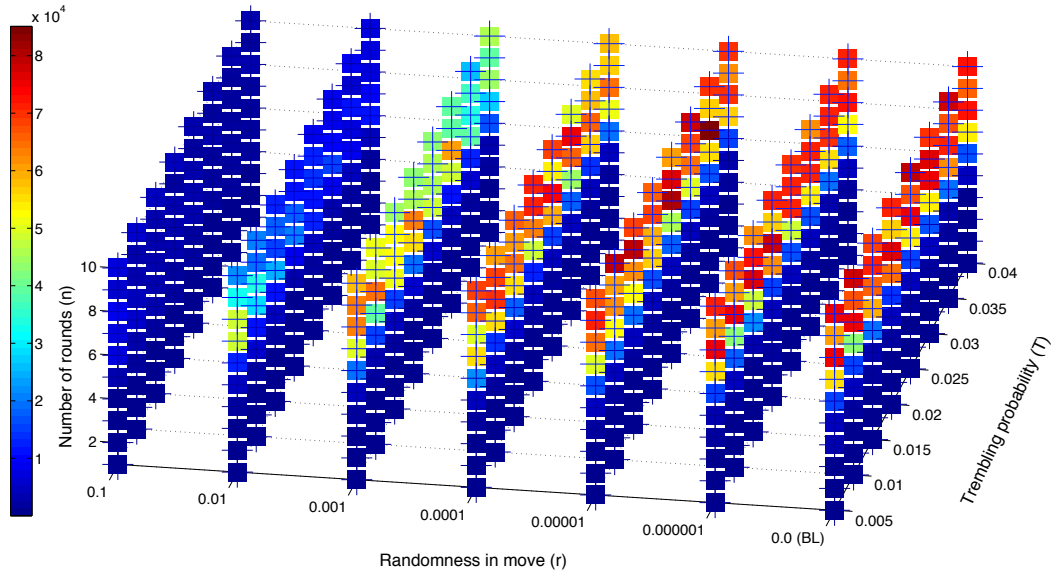


Figure B.43: *Average first hitting time*, across 50 runs for all values of r, T and n (perception errors). See Figure B.88 for the magnitudes of errors in detail.

Average stay in equilibrium

Figure B.44 shows the *average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n, for perception errors.

Average distance from equilibrium

Figure B.45 shows the *average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for perception errors.

Average payoff

Figures B.46 and B.47 show the *average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures B.48 and B.49 show the *average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

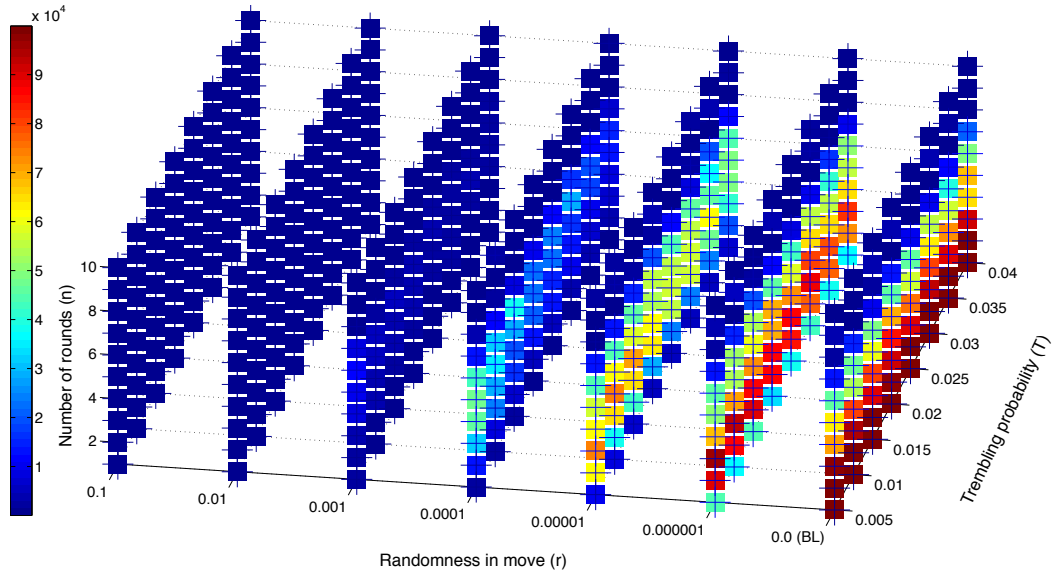


Figure B.44: *Average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n (perception errors). See Figure B.89 for the magnitudes of errors in detail.

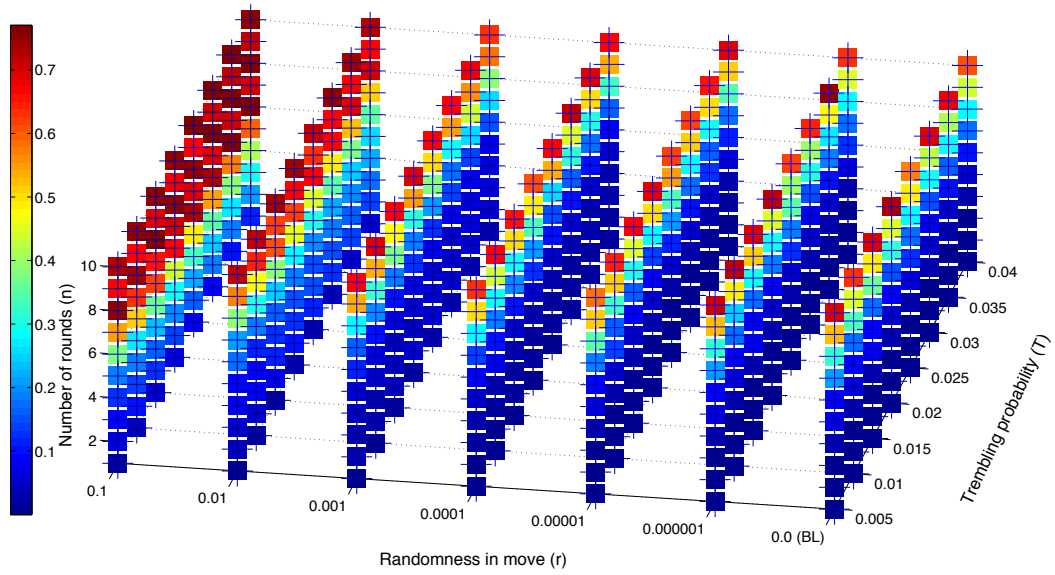


Figure B.45: *Average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (perception errors). See Figure B.90 for the magnitudes of errors in detail.

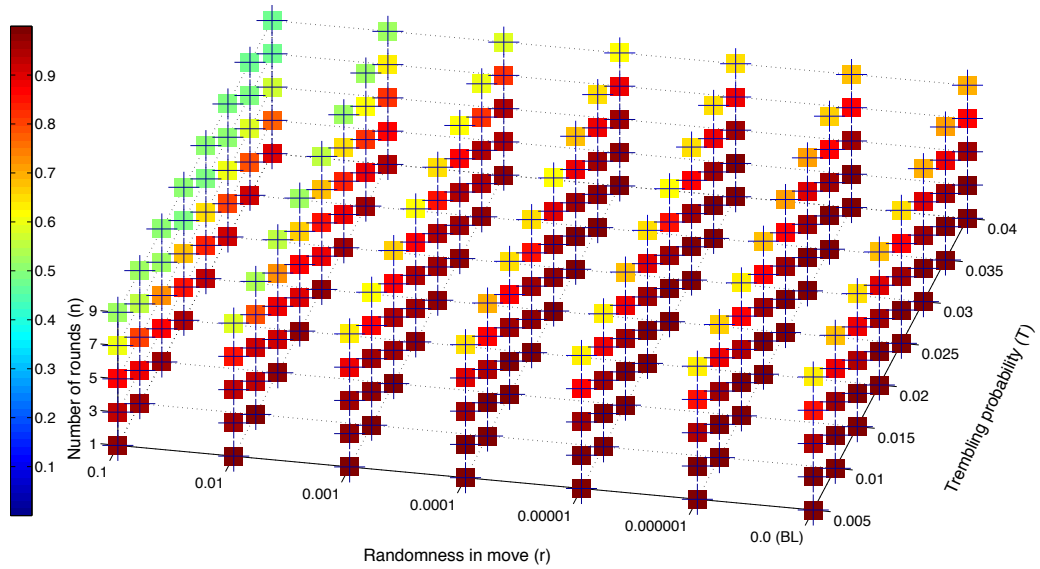


Figure B.46: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure B.91 for the magnitudes of errors in detail.

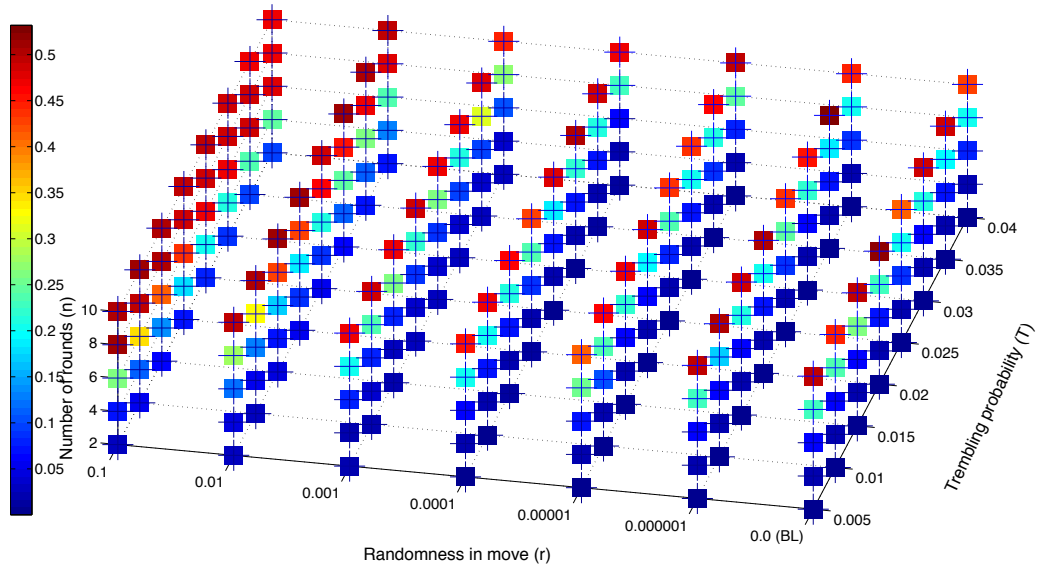


Figure B.47: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure B.92 for the magnitudes of errors in detail.

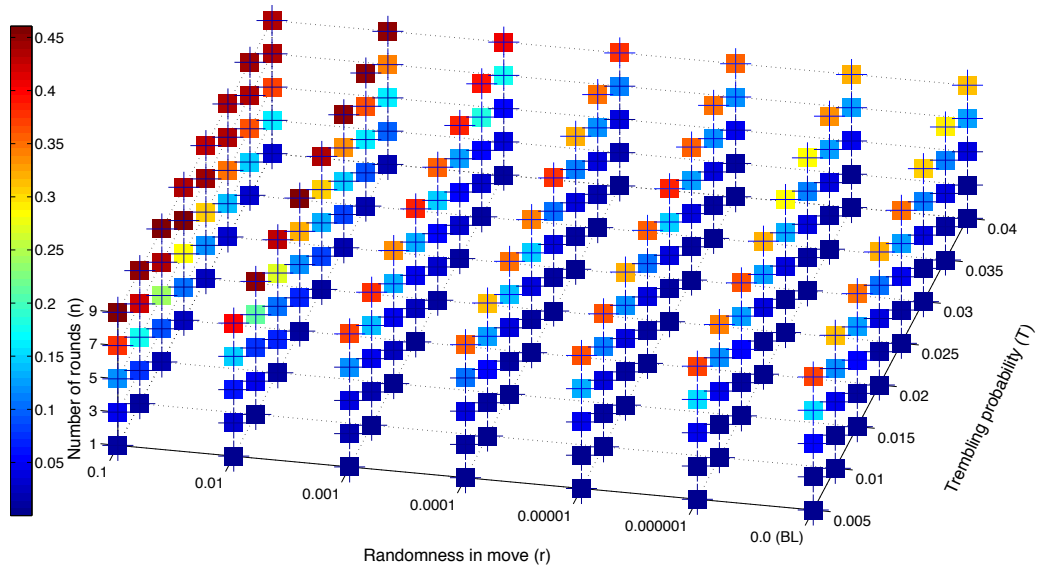


Figure B.48: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure B.93 for the magnitudes of errors in detail.

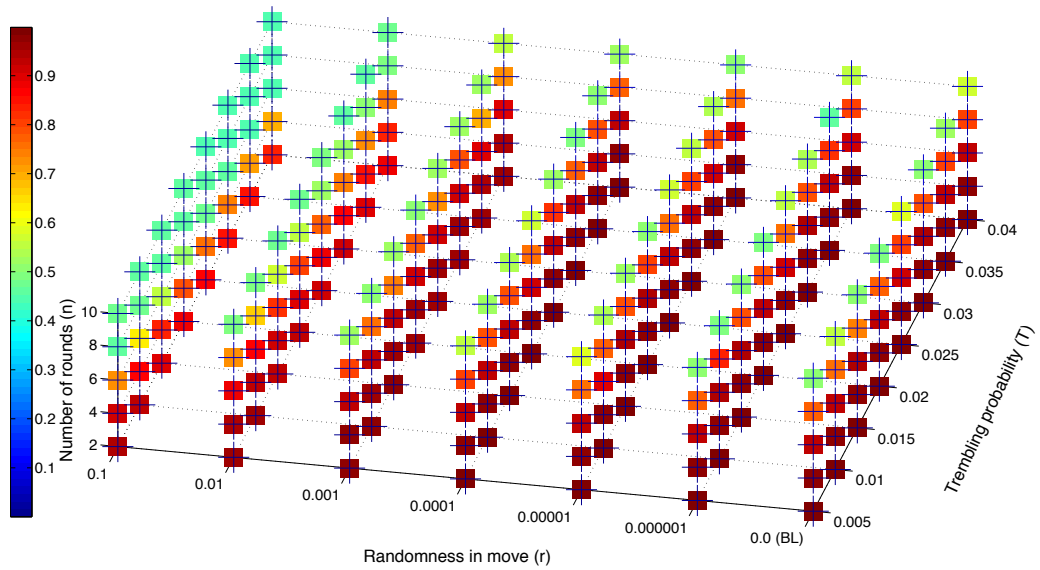


Figure B.49: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure B.94 for the magnitudes of errors in detail.

Average maximum payoff

Figures B.50 and B.51 show the *average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

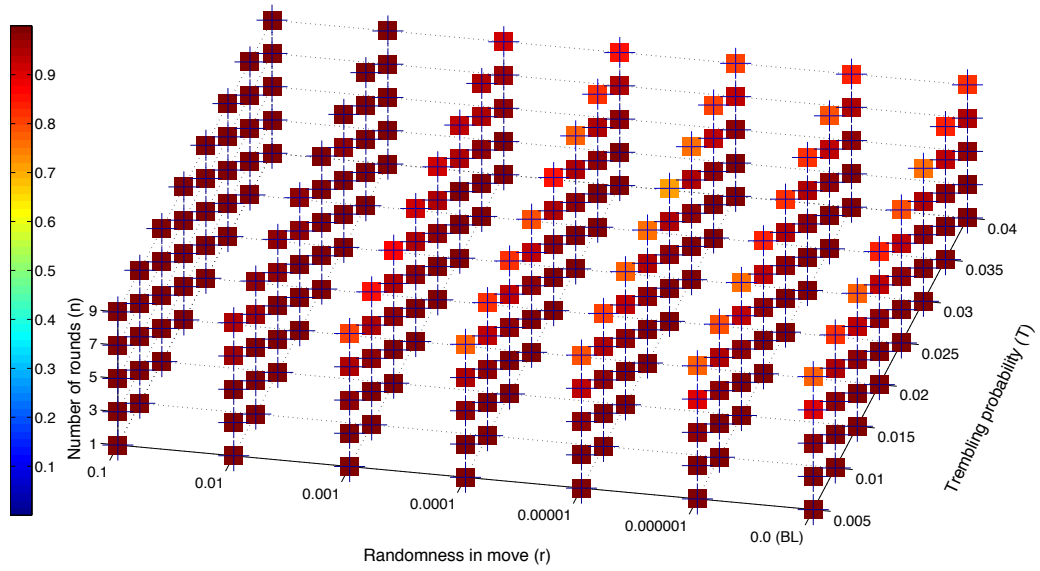


Figure B.50: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure B.95 for the magnitudes of errors in detail.

Figures B.52 and B.53 show the *average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Average minimum payoff

Figures B.54 and B.55 show the *average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures B.56 and B.57 show the *average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

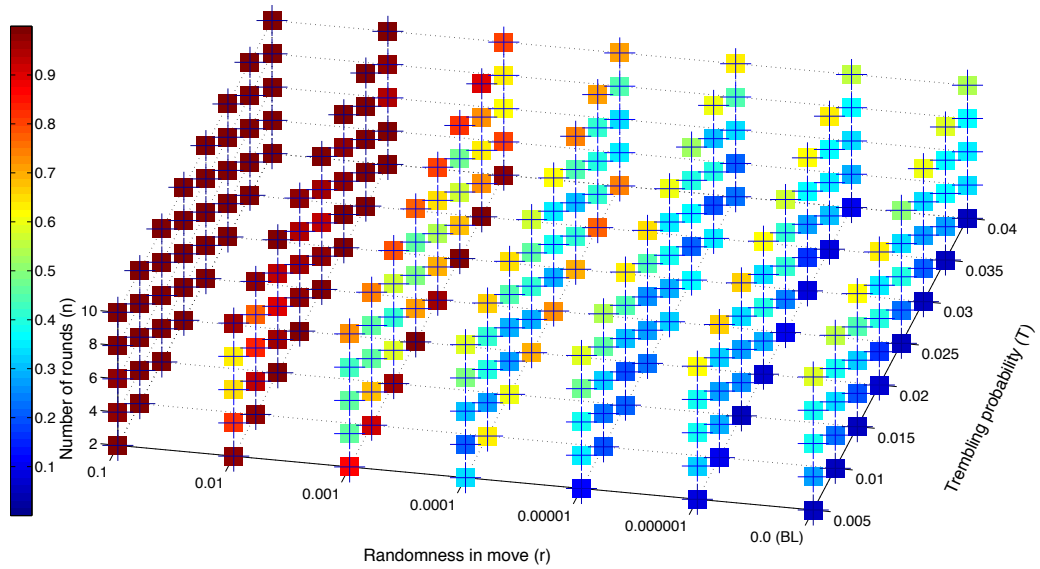


Figure B.51: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure B.96 for the magnitudes of errors in detail.

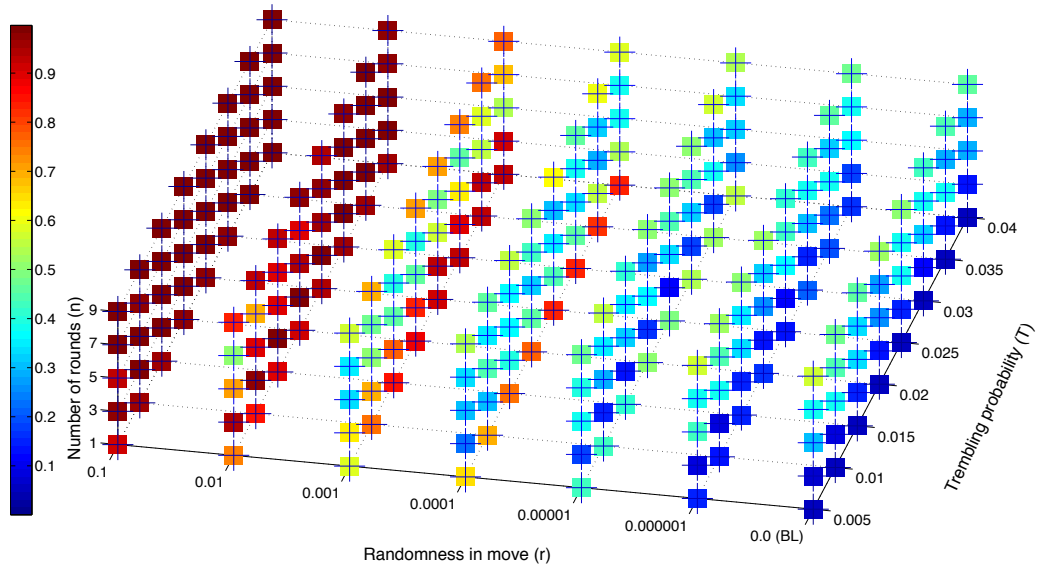


Figure B.52: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure B.97 for the magnitudes of errors in detail.

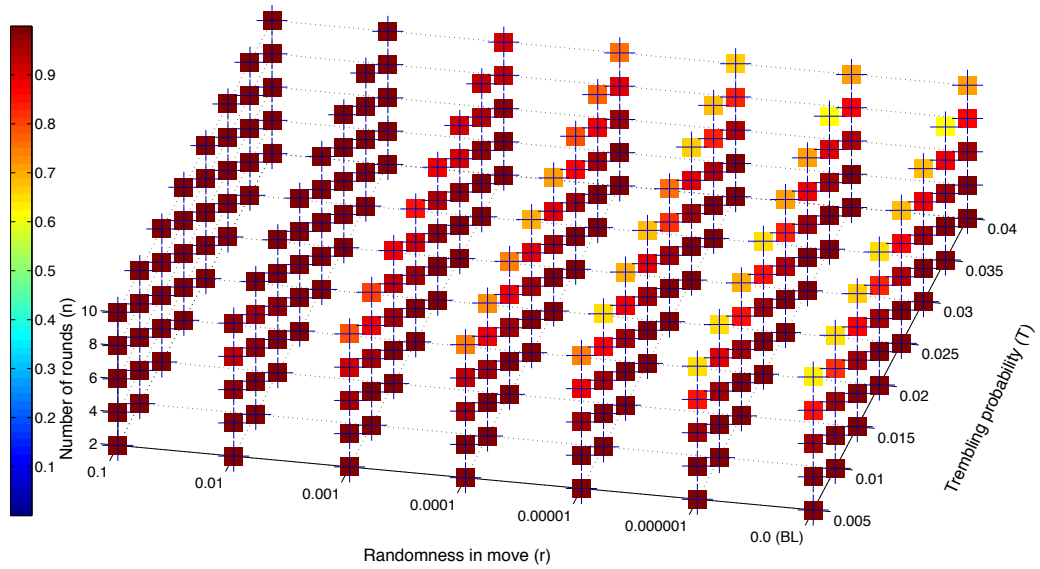


Figure B.53: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure B.98 for the magnitudes of errors in detail.

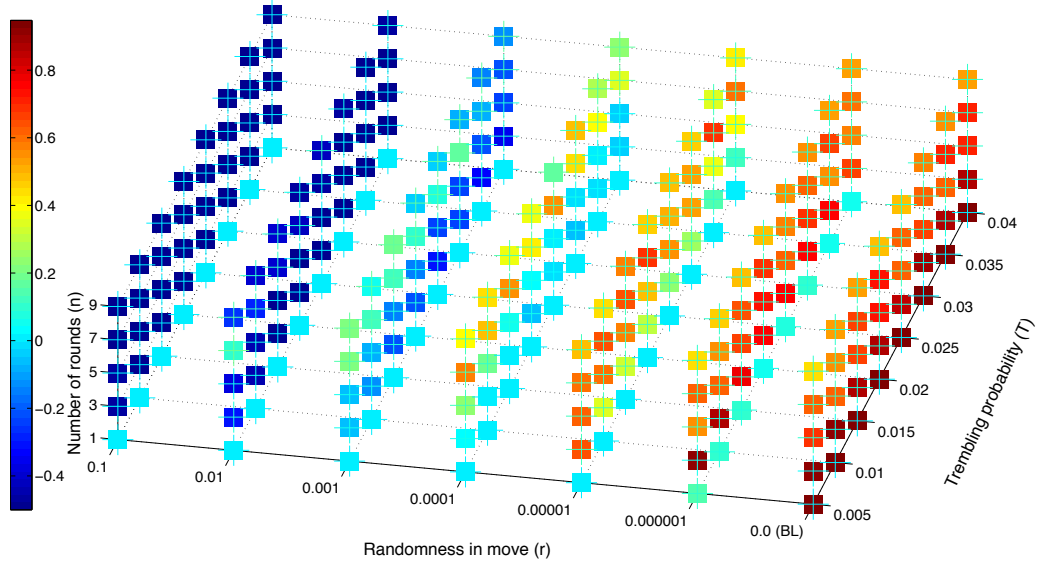


Figure B.54: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure B.99 for the magnitudes of errors in detail.

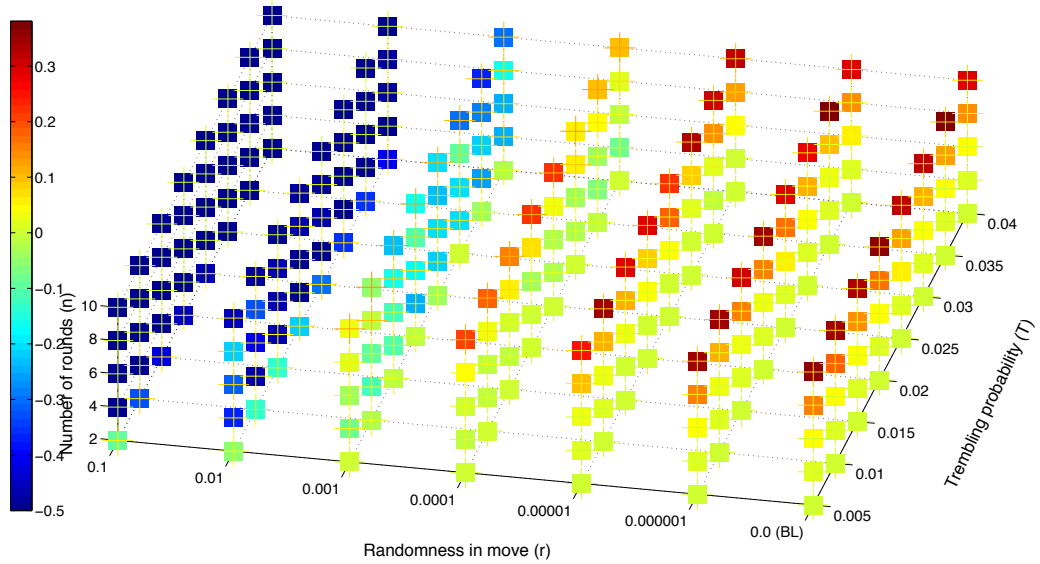


Figure B.55: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure B.100 for the magnitudes of errors in detail.

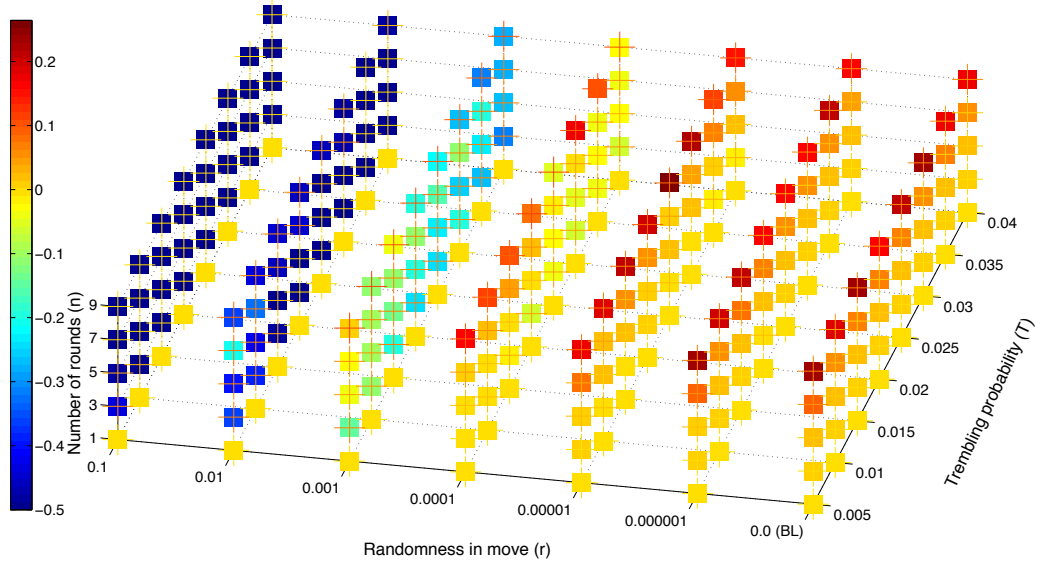


Figure B.56: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors). See Figure B.101 for the magnitudes of errors in detail.

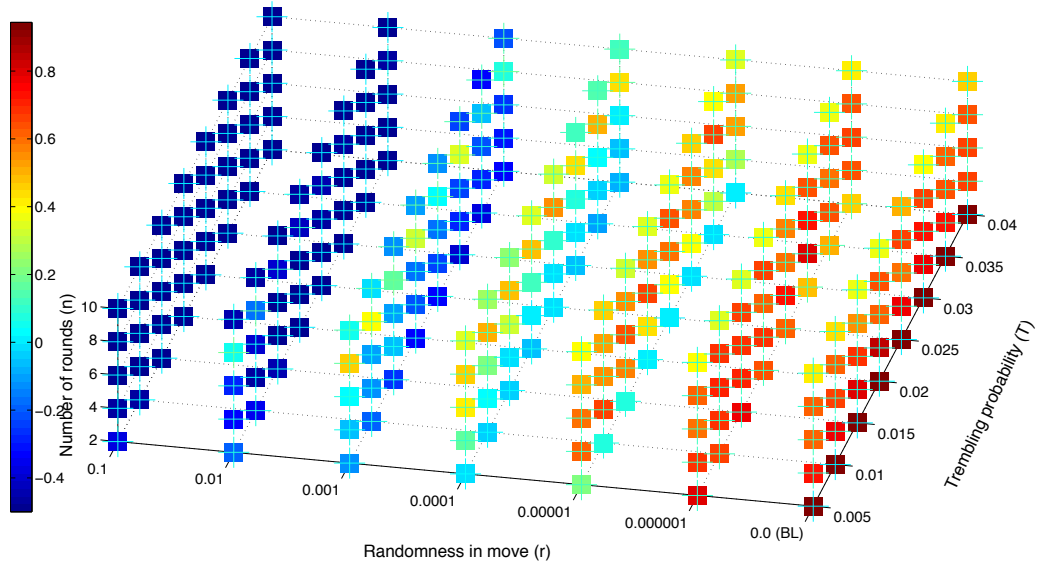


Figure B.57: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors). See Figure B.102 for the magnitudes of errors in detail.

B.2.3 Implementation and Perception Errors

Average first hitting time

Figure B.58 shows the *average first hitting time*, across 50 runs for all values of r, T and n, for implementation and perception errors.

Average stay in equilibrium

Figure B.59 shows the *average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n, for implementation and perception errors.

Average distance from equilibrium

Figure B.60 shows the *average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for implementation and perception errors.

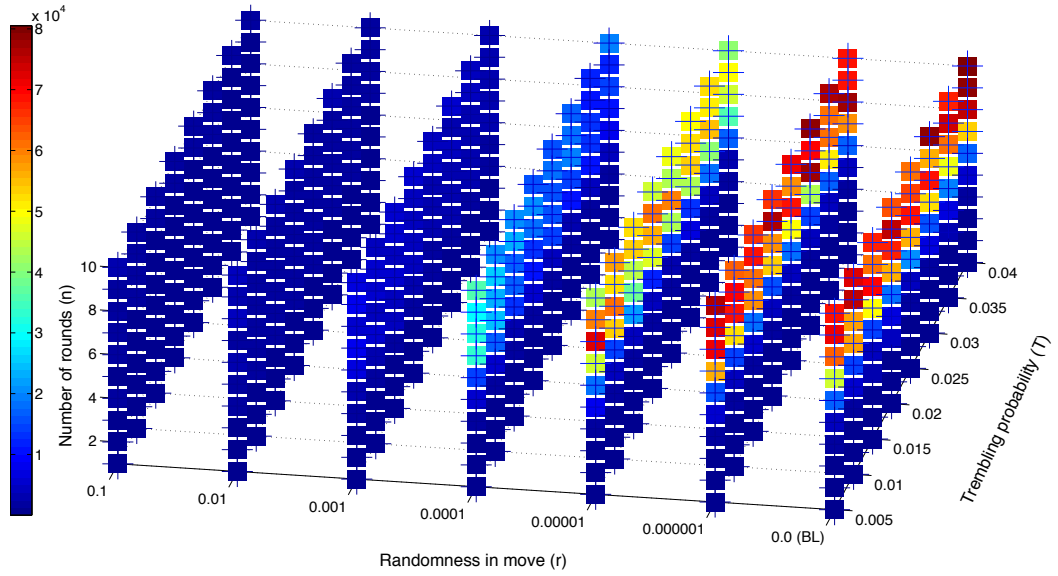


Figure B.58: *Average first hitting time*, across 50 runs for all values of r, T and n (implementation and perception errors). See Figure B.103 for the magnitudes of errors in detail.

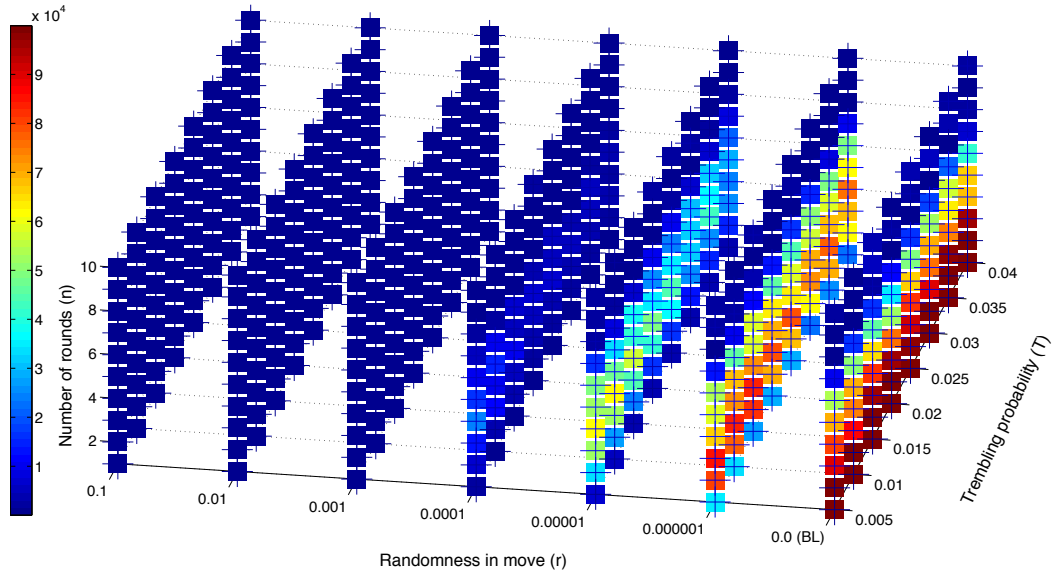


Figure B.59: *Average stay within ϵ distance from equilibrium*, across 50 runs for all values of r, T and n (implementation and perception errors). See Figure B.104 for the magnitudes of errors in detail.

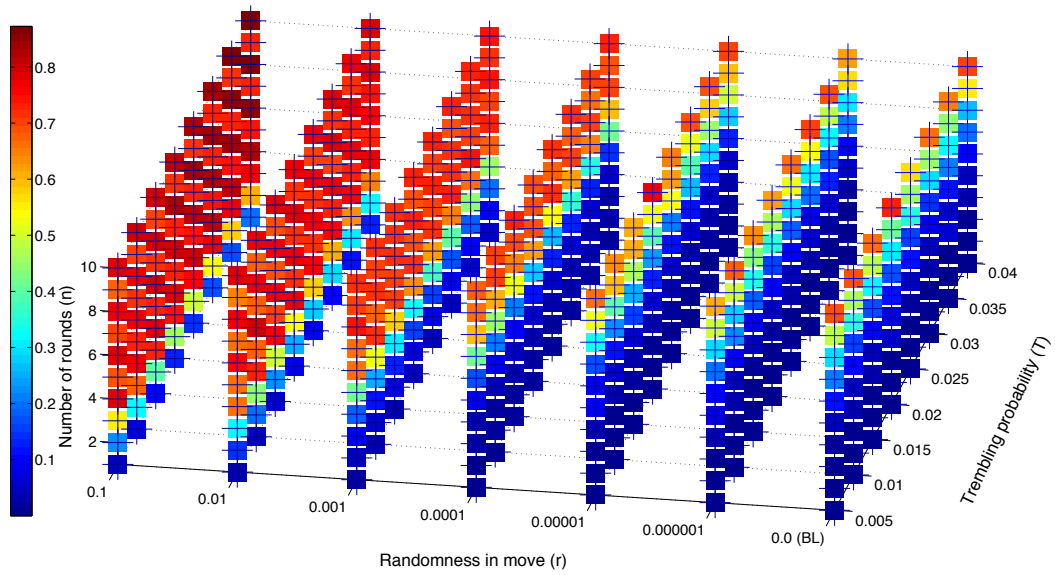


Figure B.60: *Average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (implementation and perception errors). See Figure B.105 for the magnitudes of errors in detail.

Average payoff

Figures B.61 and B.62 show the *average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Figures B.63 and B.64 show the *average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Average maximum payoff

Figures B.65 and B.66 show the *average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Figures B.67 and B.68 show the *average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

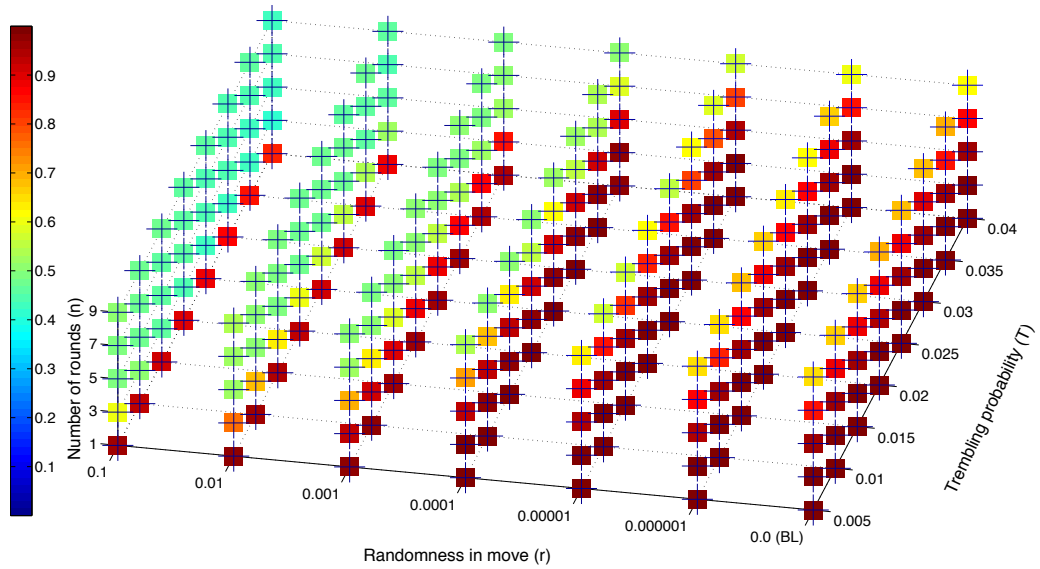


Figure B.61: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure B.106 for the magnitudes of errors in detail.

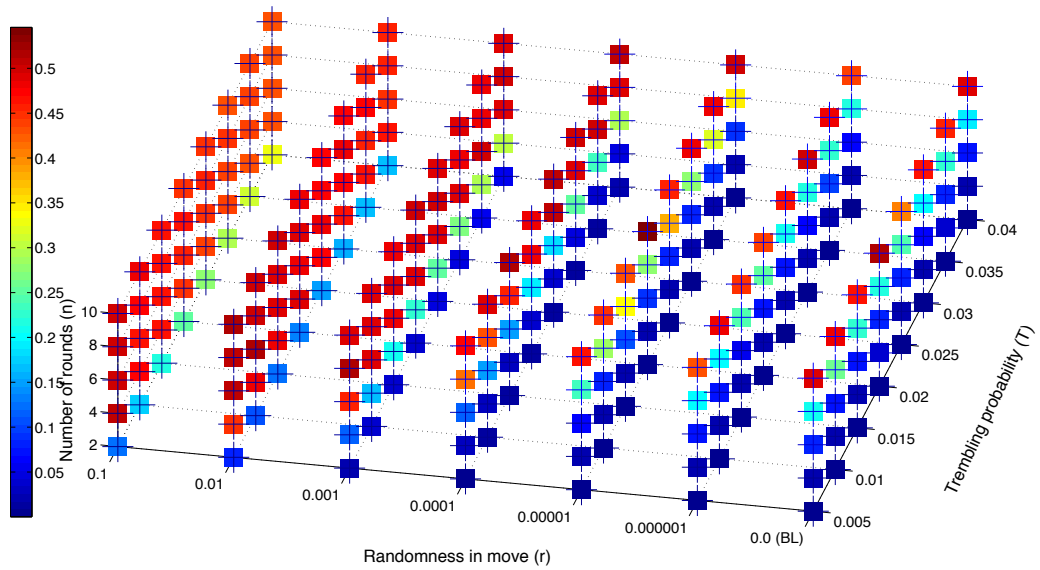


Figure B.62: *Average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure B.107 for the magnitudes of errors in detail.

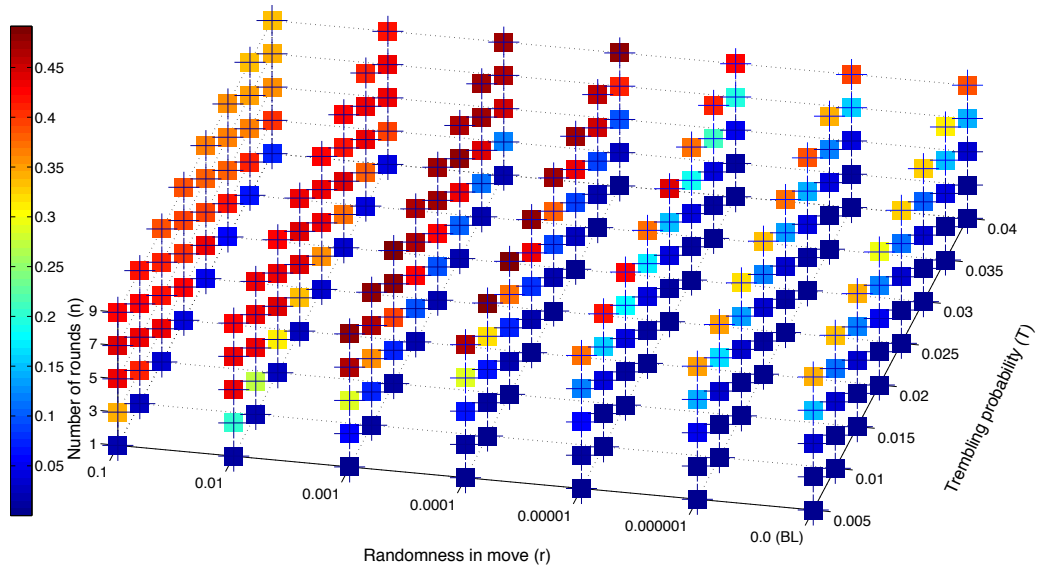


Figure B.63: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure B.108 for the magnitudes of errors in detail.

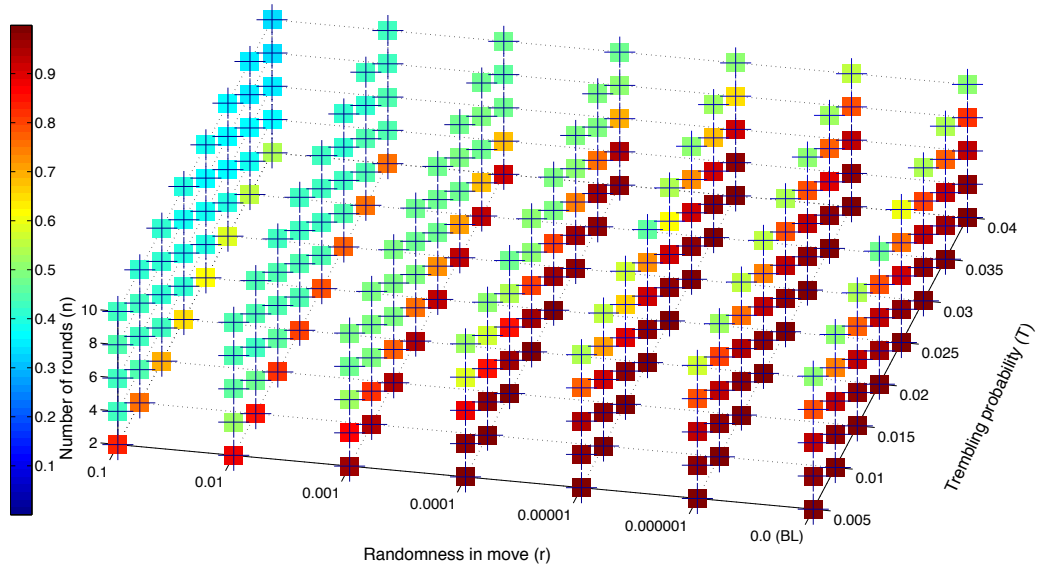


Figure B.64: *Average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure B.109 for the magnitudes of errors in detail.

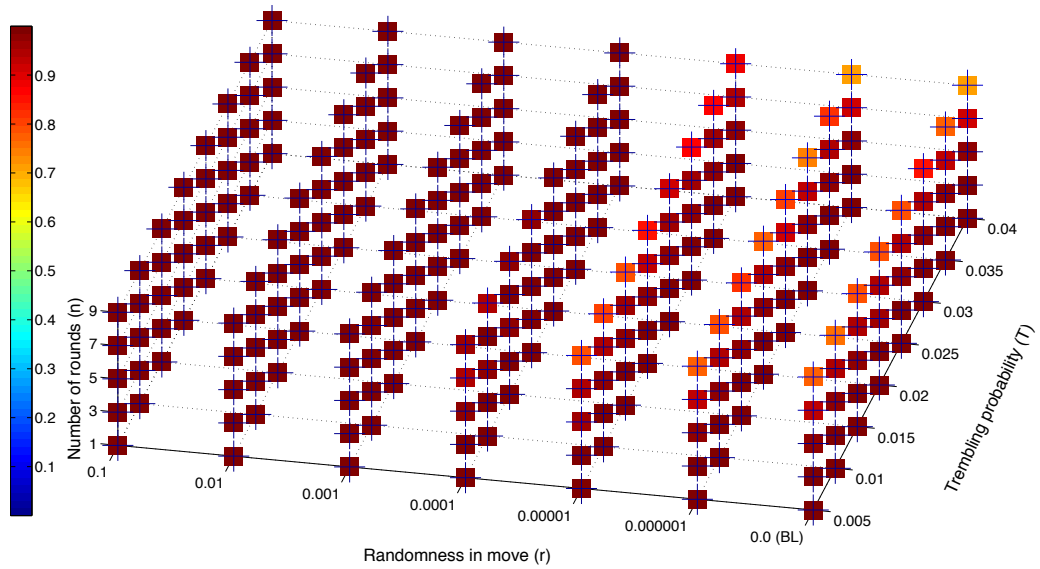


Figure B.65: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure B.110 for the magnitudes of errors in detail.

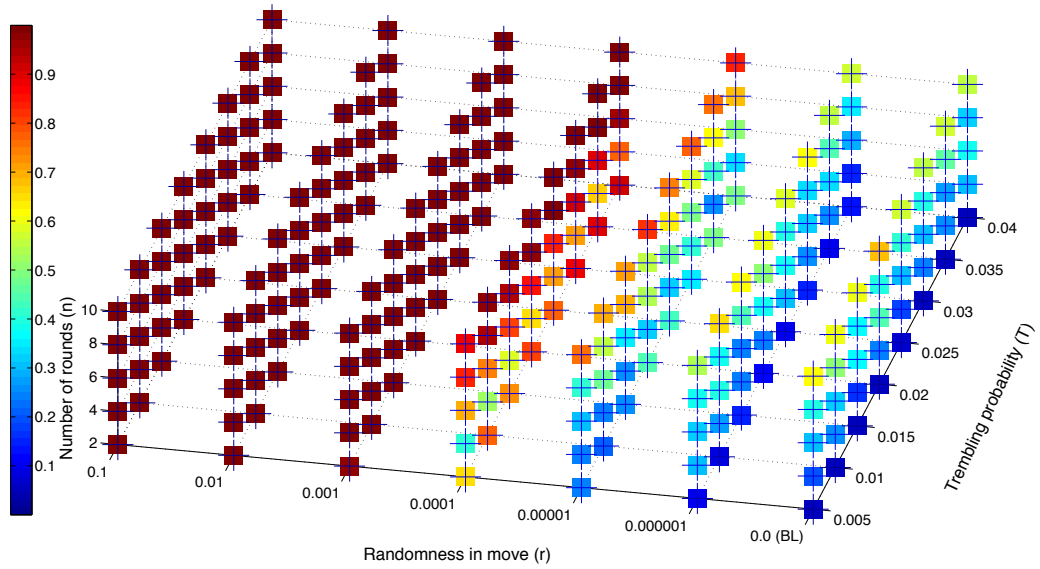


Figure B.66: *Average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure B.111 for the magnitudes of errors in detail.

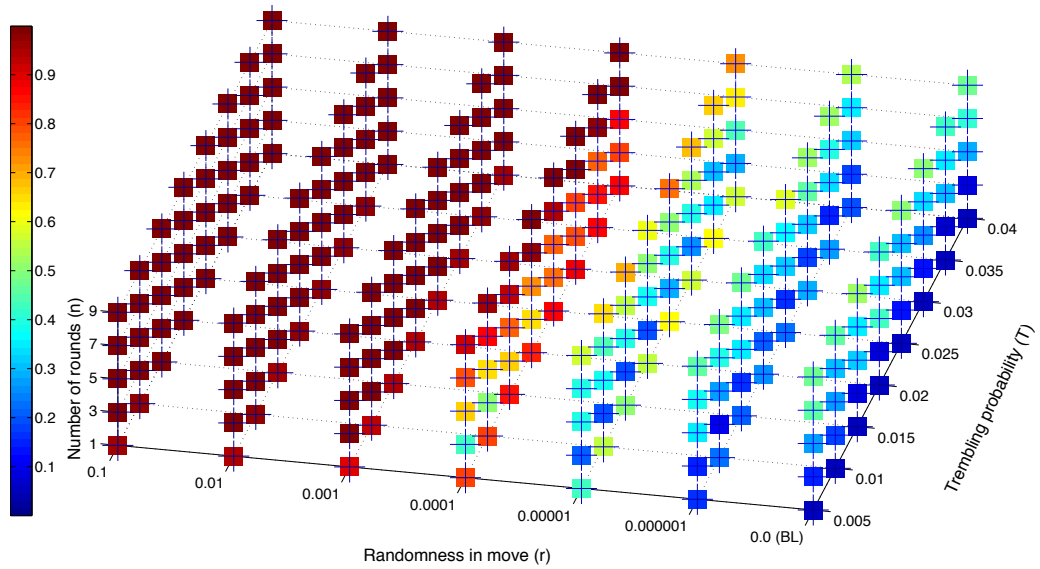


Figure B.67: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure B.112 for the magnitudes of errors in detail.

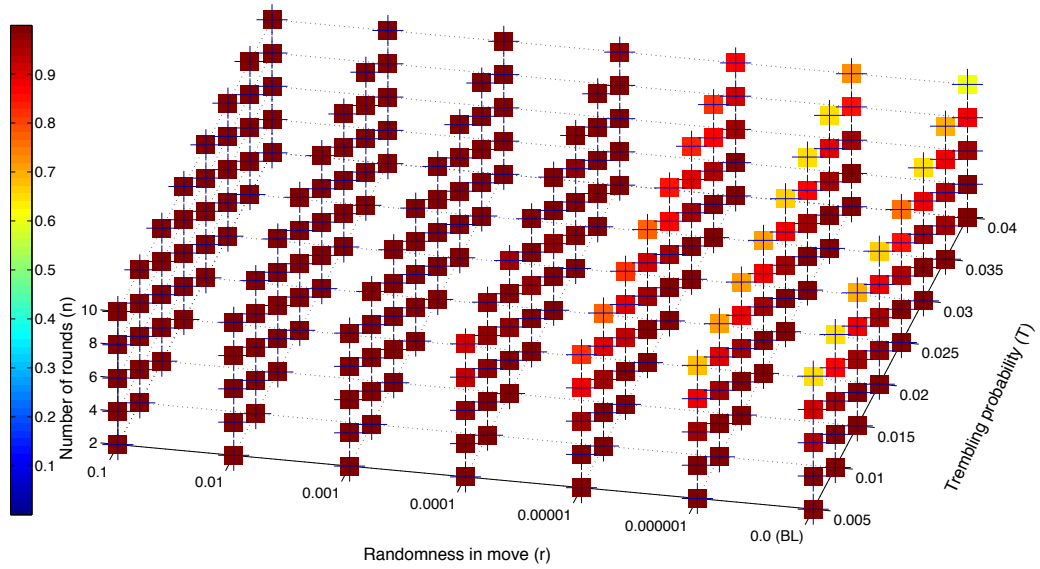


Figure B.68: *Average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure B.113 for the magnitudes of errors in detail.

Average minimum payoff

Figures B.69 and B.70 show the *average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

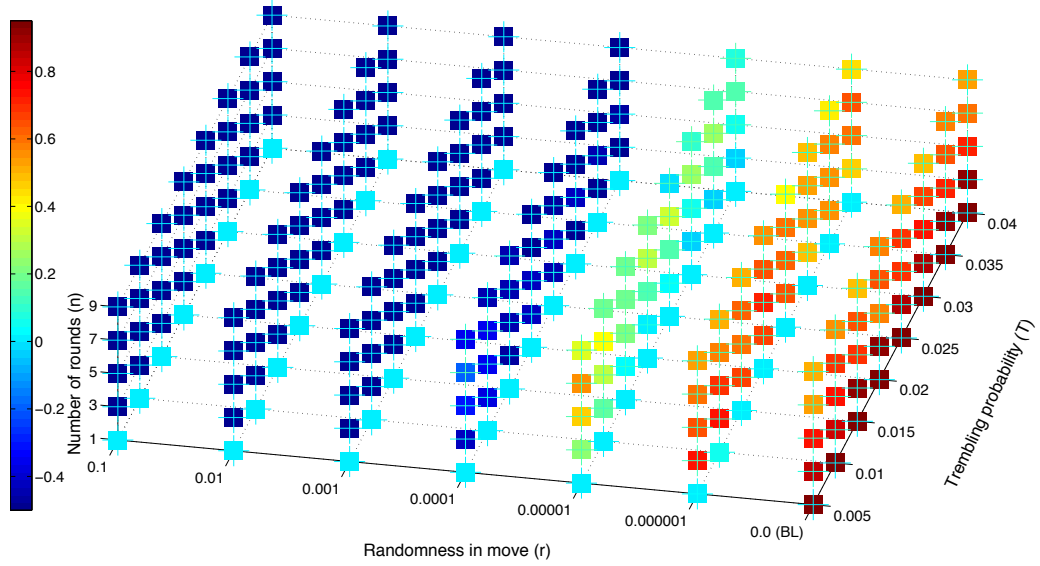


Figure B.69: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure B.114 for the magnitudes of errors in detail.

Figures B.71 and B.72 show the *average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

B.3 Confidence Across 50 Runs

For each type of bounded rationality, we plot the standard error (95% confidence) for each evaluation metric, for each game and rationality bound setting. This is in the same order as the graphs in Section B.2 for easy reference.

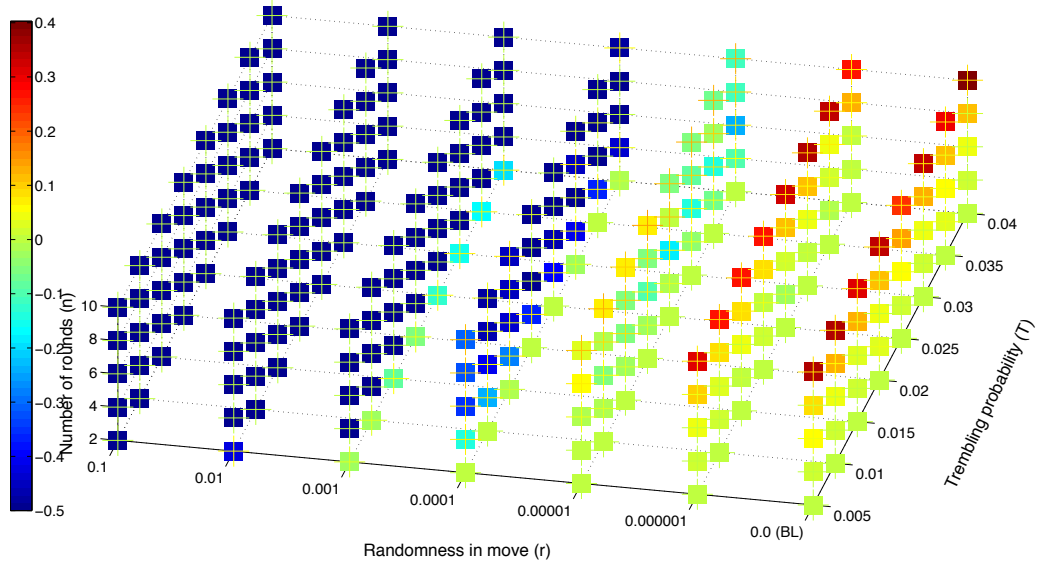


Figure B.70: *Average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure B.115 for the magnitudes of errors in detail.

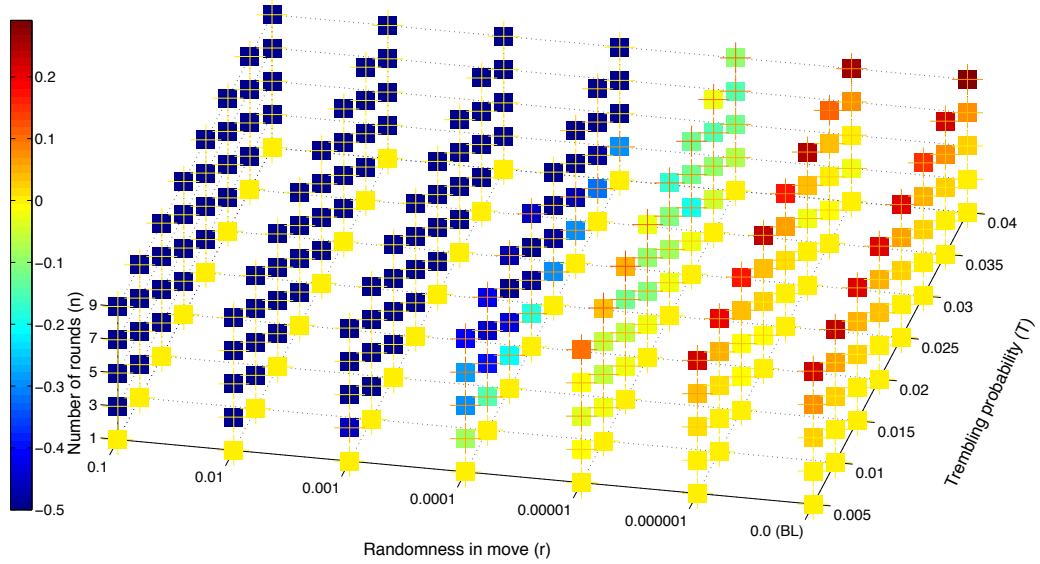


Figure B.71: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors). See Figure B.116 for the magnitudes of errors in detail.

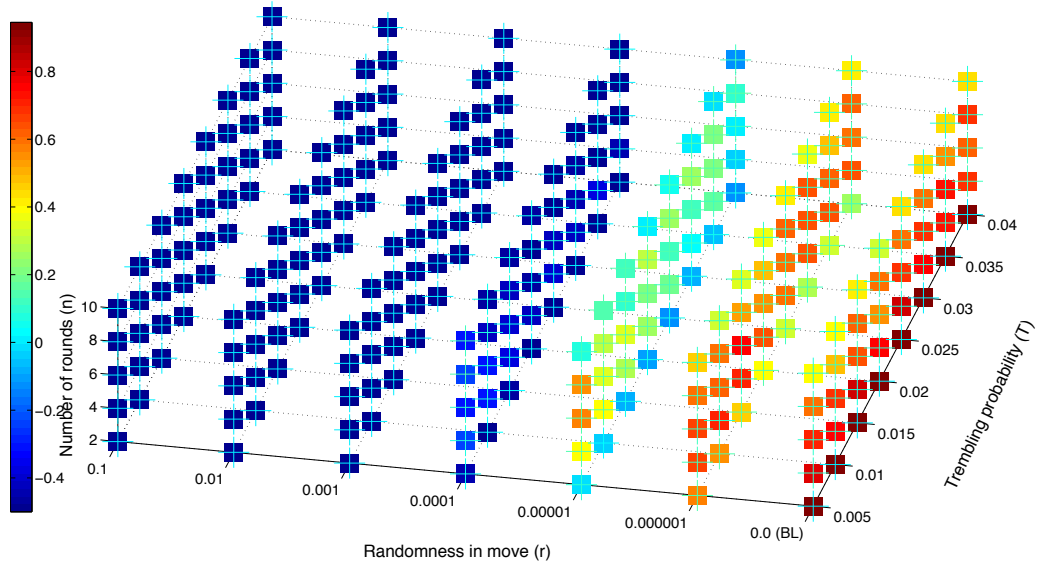


Figure B.72: *Average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors). See Figure B.117 for the magnitudes of errors in detail.

B.3.1 Implementation Errors

Error at 95% confidence for average first hitting time

Figure B.73 shows the *error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n, for implementation errors.

Error at 95% confidence for average stay in equilibrium

Figure B.74 shows the *error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n, for implementation errors.

Error at 95% confidence for average distance from equilibrium

Figure B.75 shows the *error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for implementation errors.

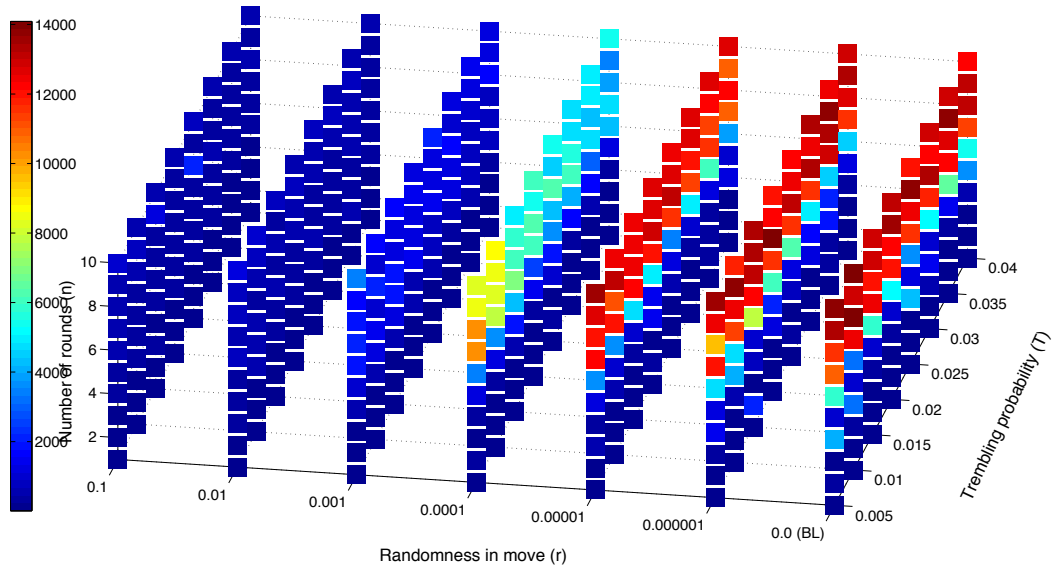


Figure B.73: *Error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n (implementation errors).

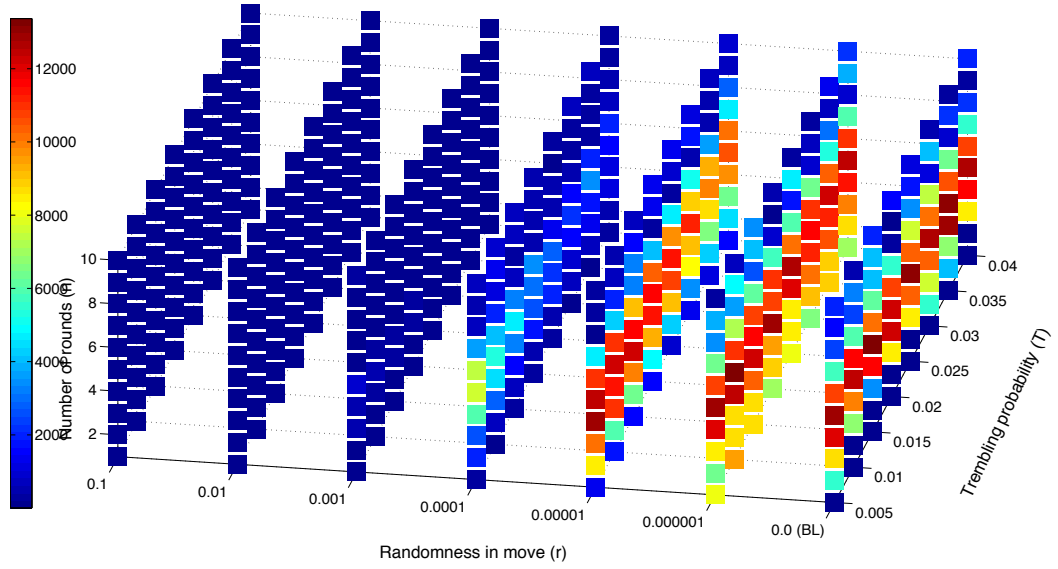


Figure B.74: *Error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n (implementation errors).

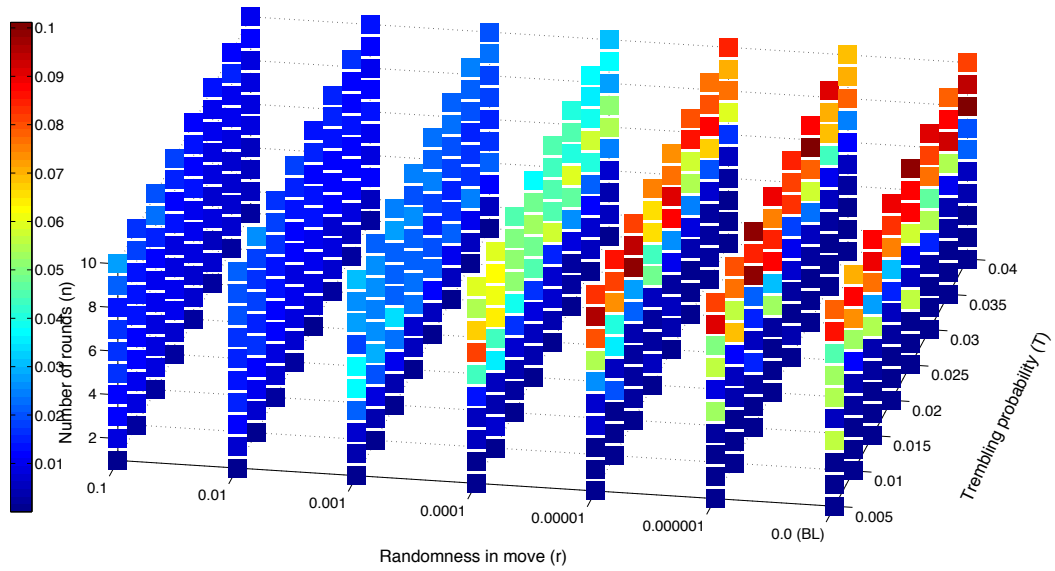


Figure B.75: *Error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (implementation errors).

Error at 95% confidence for average payoff

Figures B.76 and B.77 show the *error at* 95% *confidence for average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Figures B.78 and B.79 show the *error at* 95% *confidence for average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Error at 95% confidence for average maximum payoff

Figures B.80 and B.81 show the *error at* 95% *confidence for average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

Figures B.82 and B.83 show the *error at* 95% *confidence for average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

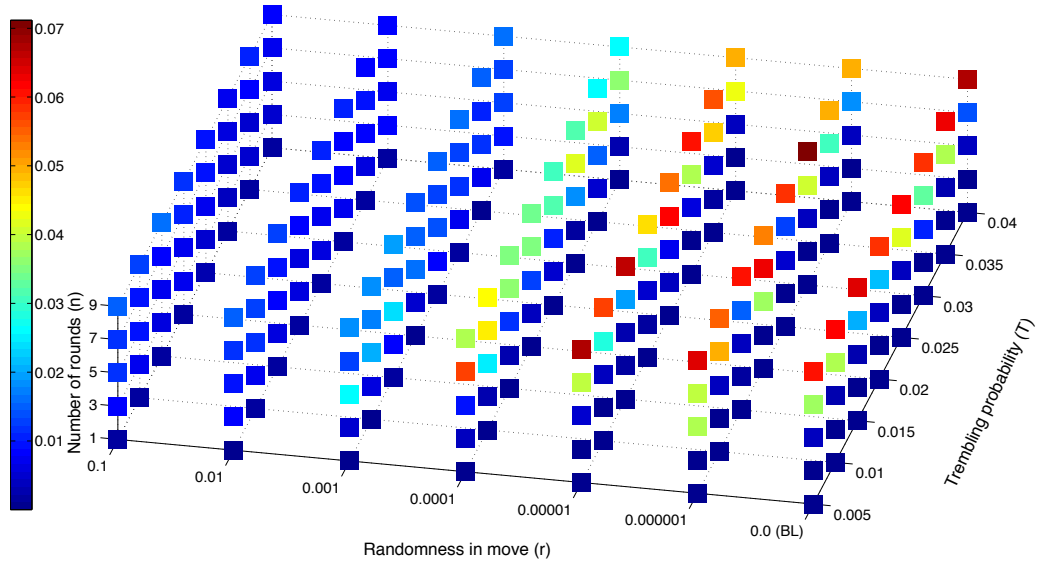


Figure B.76: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

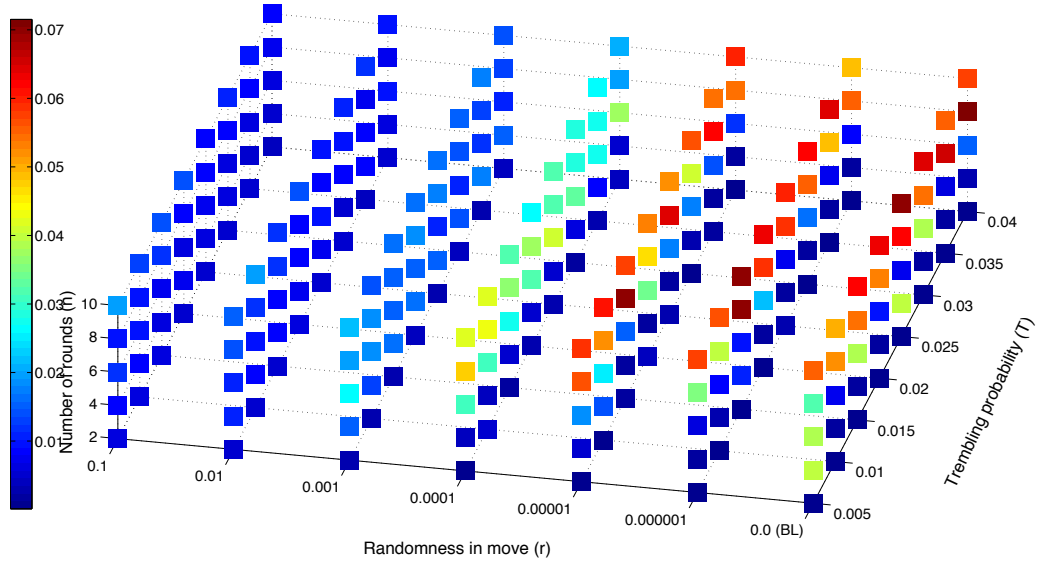


Figure B.77: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

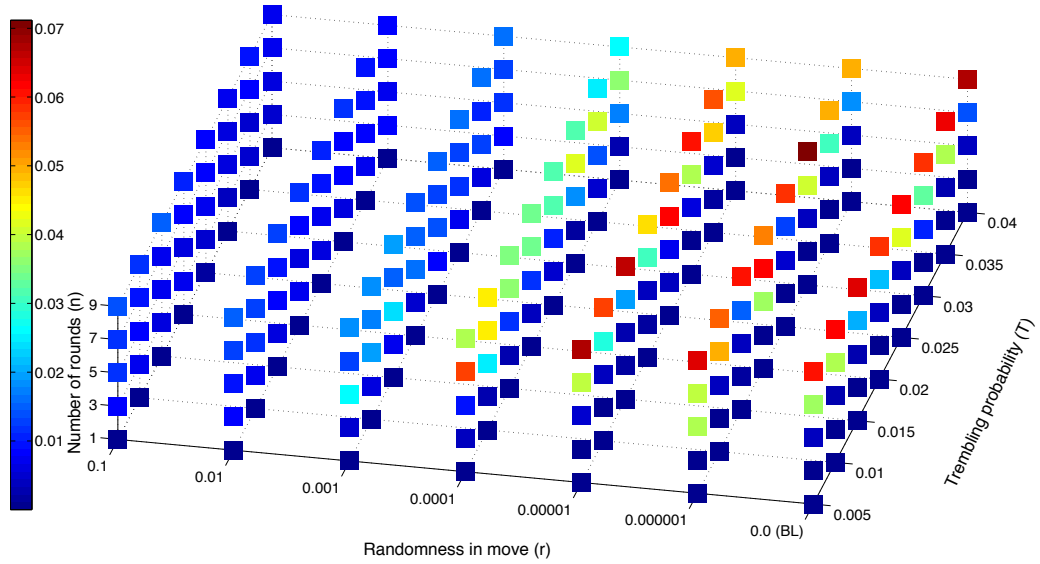


Figure B.78: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

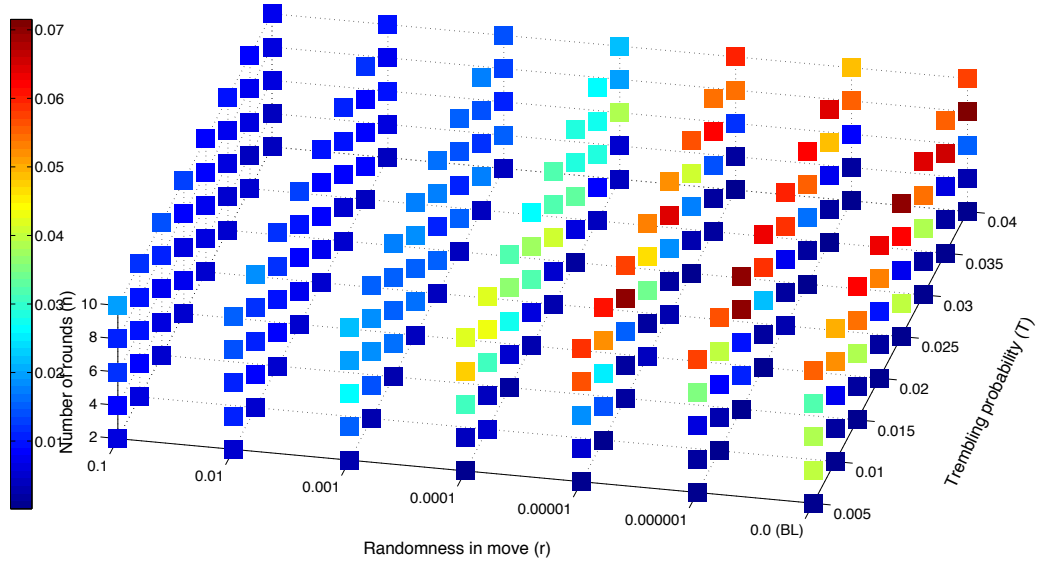


Figure B.79: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

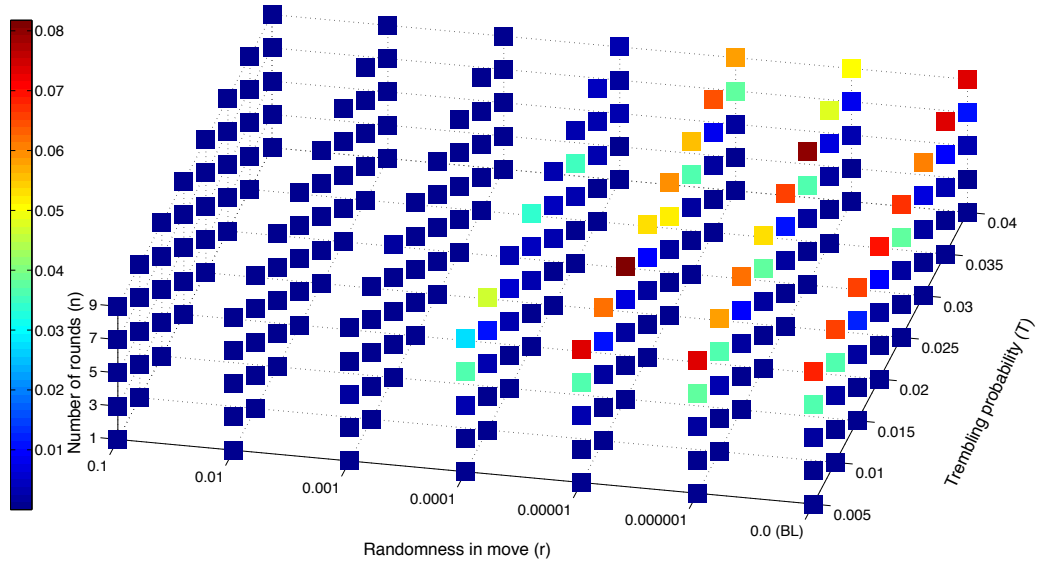


Figure B.80: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

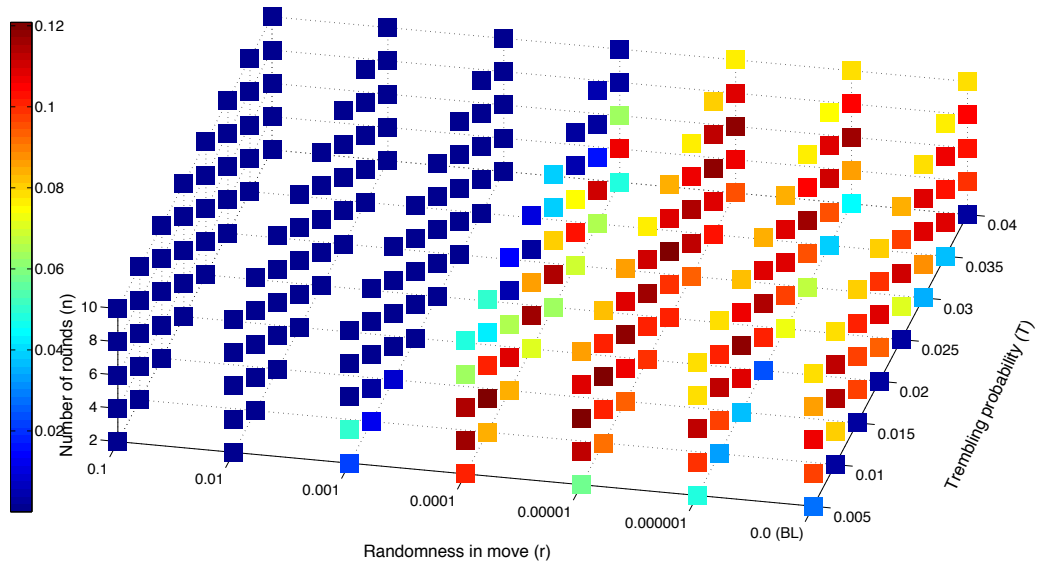


Figure B.81: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

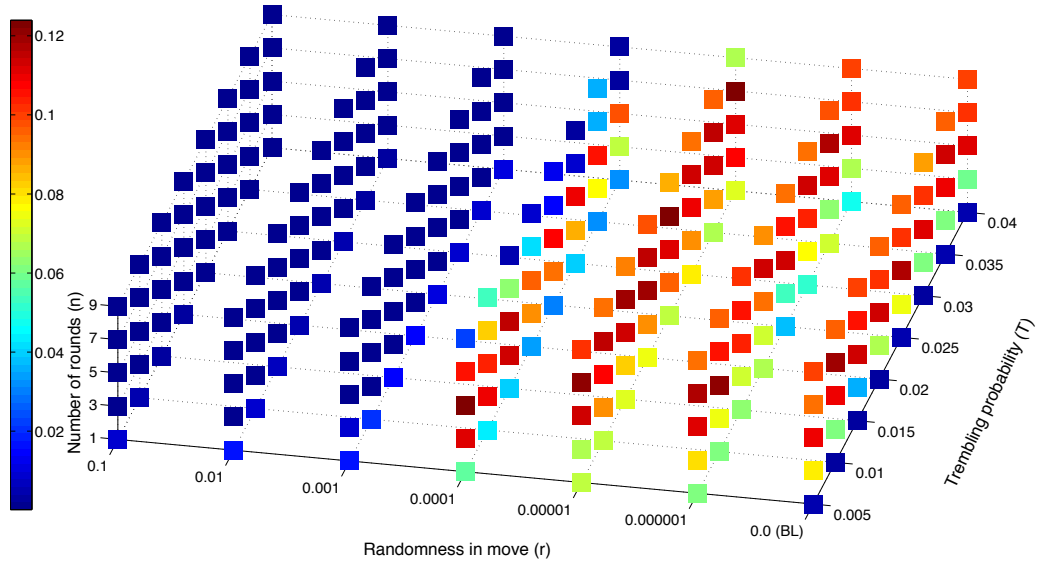


Figure B.82: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

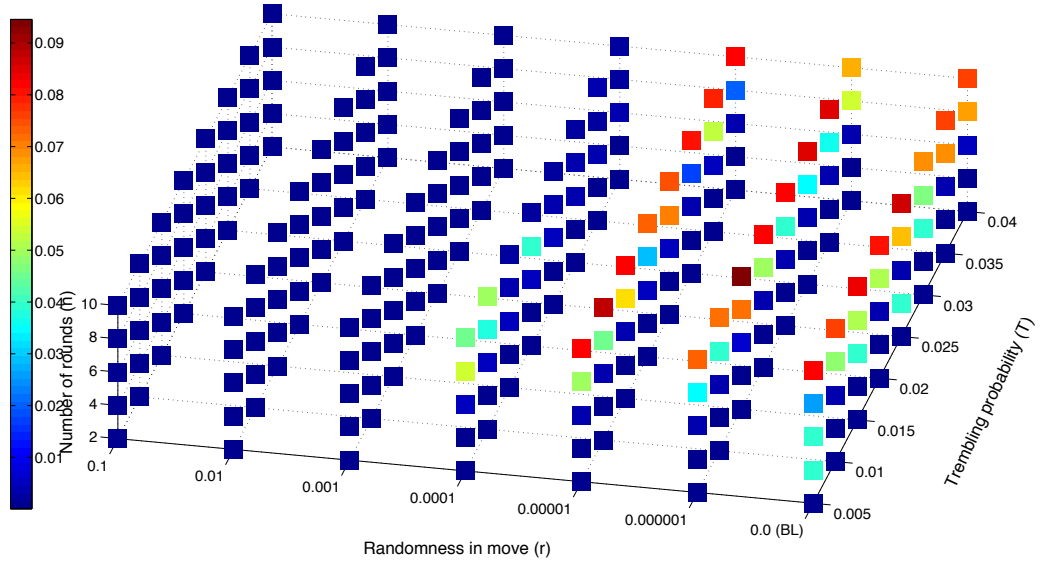


Figure B.83: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

Error at 95% confidence for average minimum payoff

Figures B.84 and B.85 show the *error at* 95% *confidence for average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

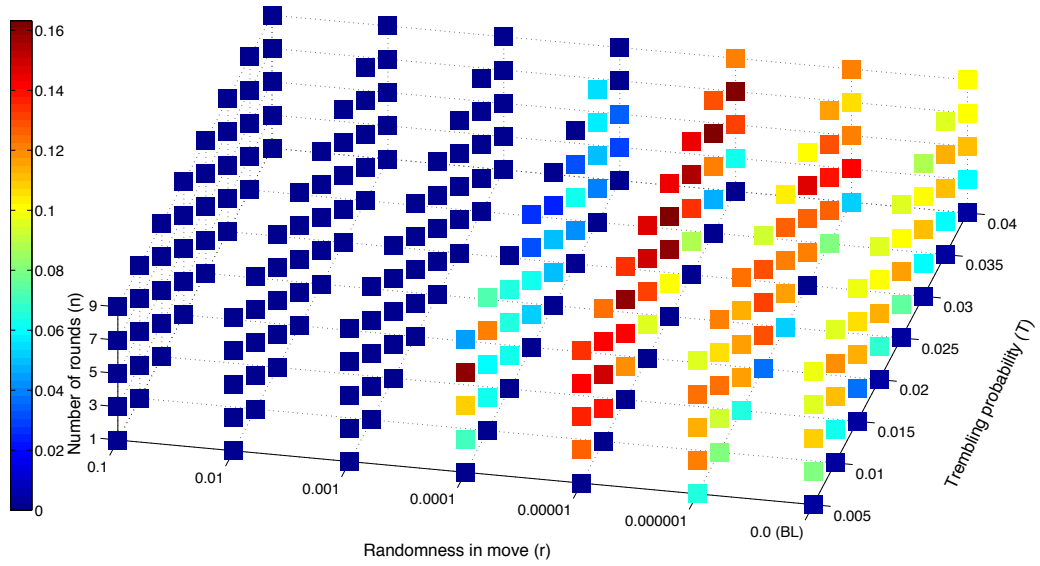


Figure B.84: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

Figures B.86 and B.87 show the *error at* 95% *confidence for average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation errors.

B.3.2 Perception Errors

Error at 95% confidence for average first hitting time

Figure B.88 shows the *error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n, for perception errors.

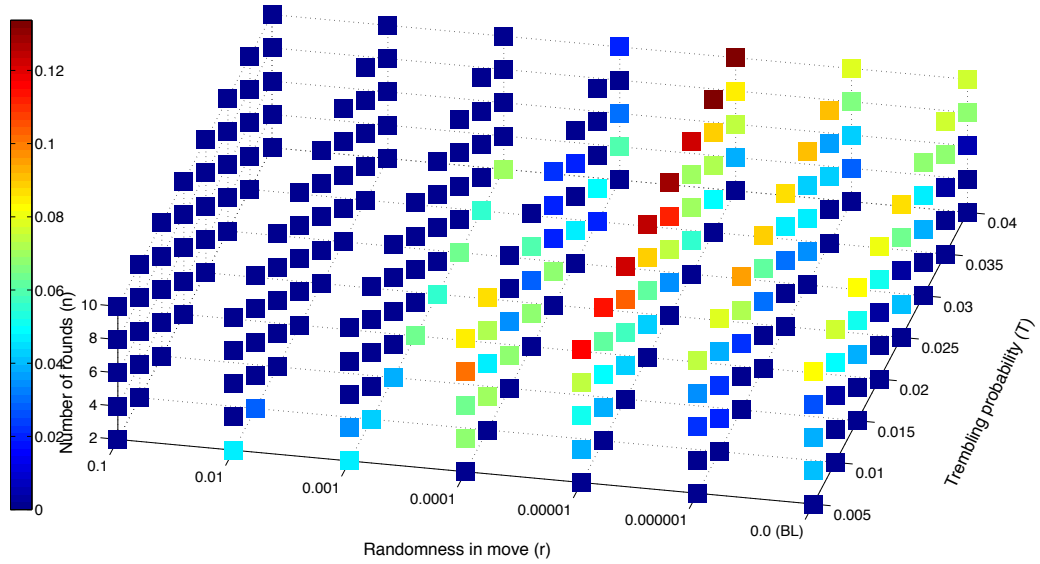


Figure B.85: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

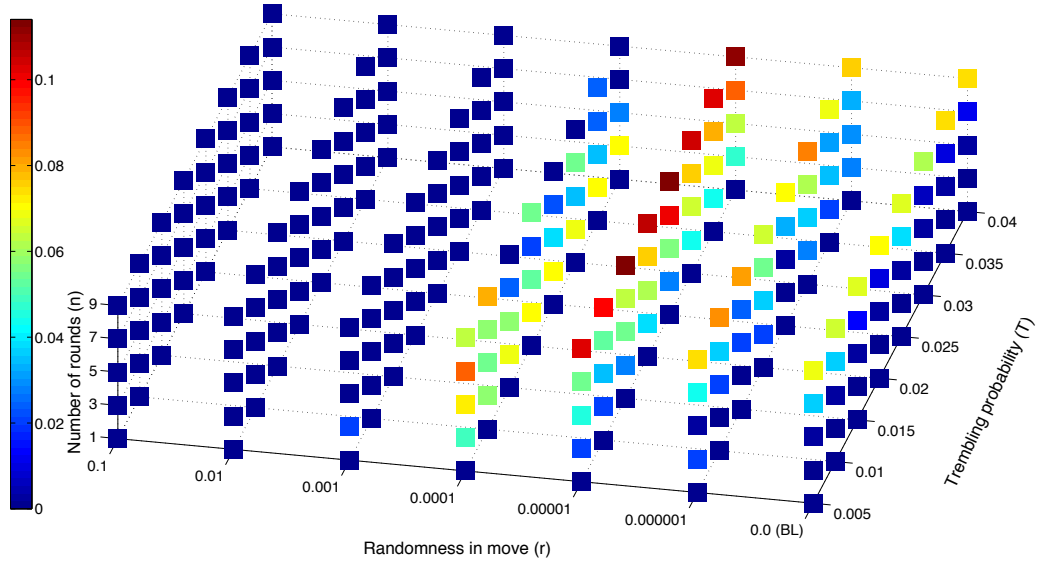


Figure B.86: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation errors).

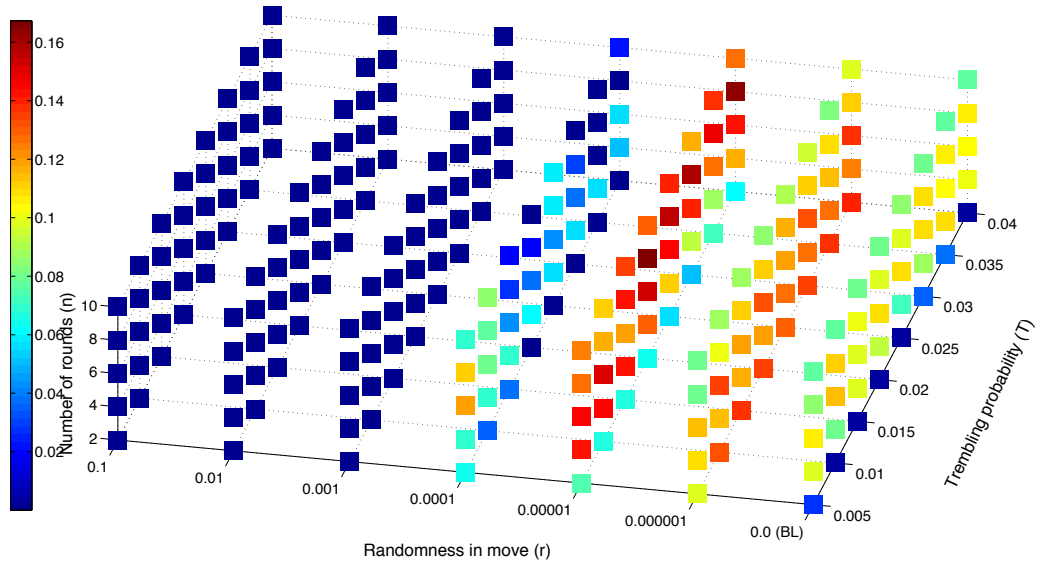


Figure B.87: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation errors).

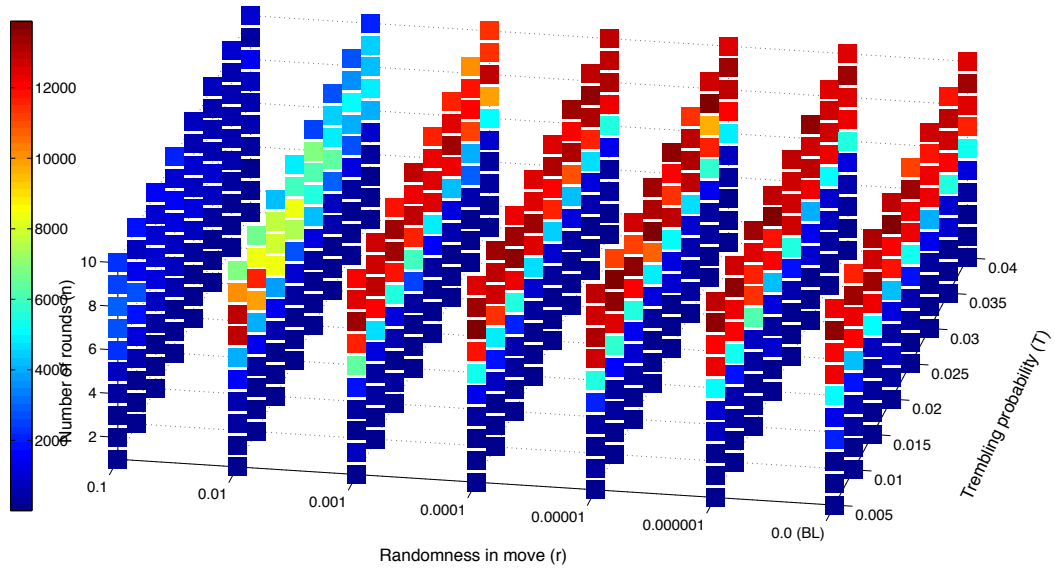


Figure B.88: *Error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n (perception errors).

Error at 95% confidence for average stay in equilibrium

Figure B.89 shows the *error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n, for perception errors.

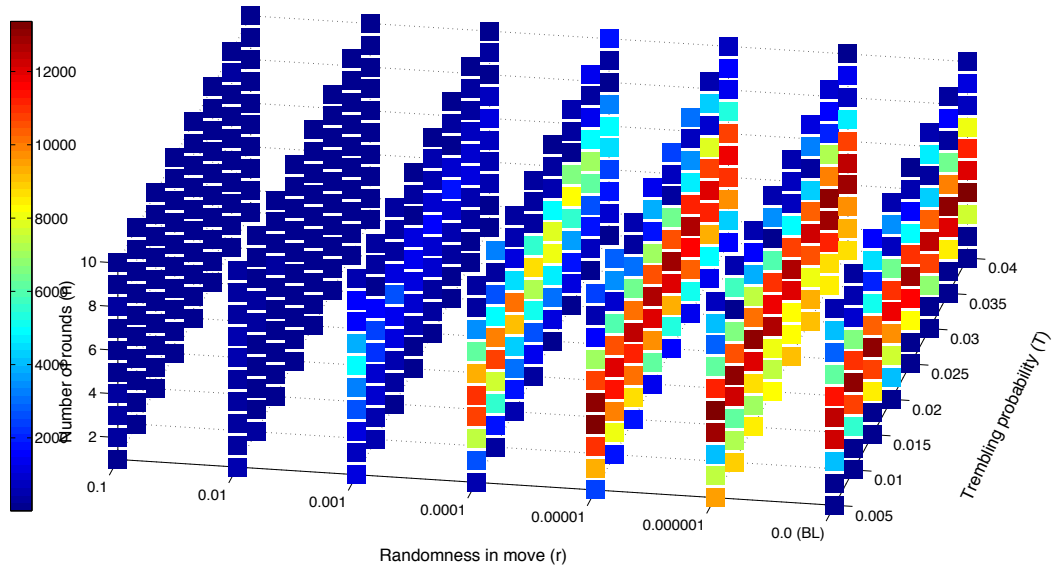


Figure B.89: *Error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n (perception errors).

Error at 95% confidence for average distance from equilibrium

Figure B.90 shows the *error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for perception errors.

Error at 95% confidence for average payoff

Figures B.91 and B.92 show the *error at* 95% *confidence for average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures B.93 and B.94 show the *error at* 95% *confidence for average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being

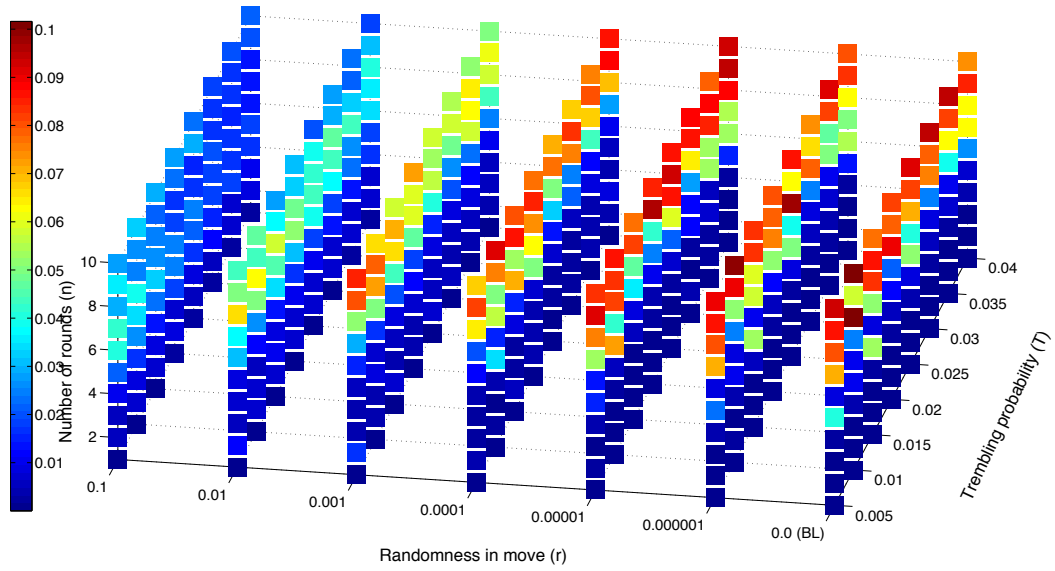


Figure B.90: *Error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (perception errors).

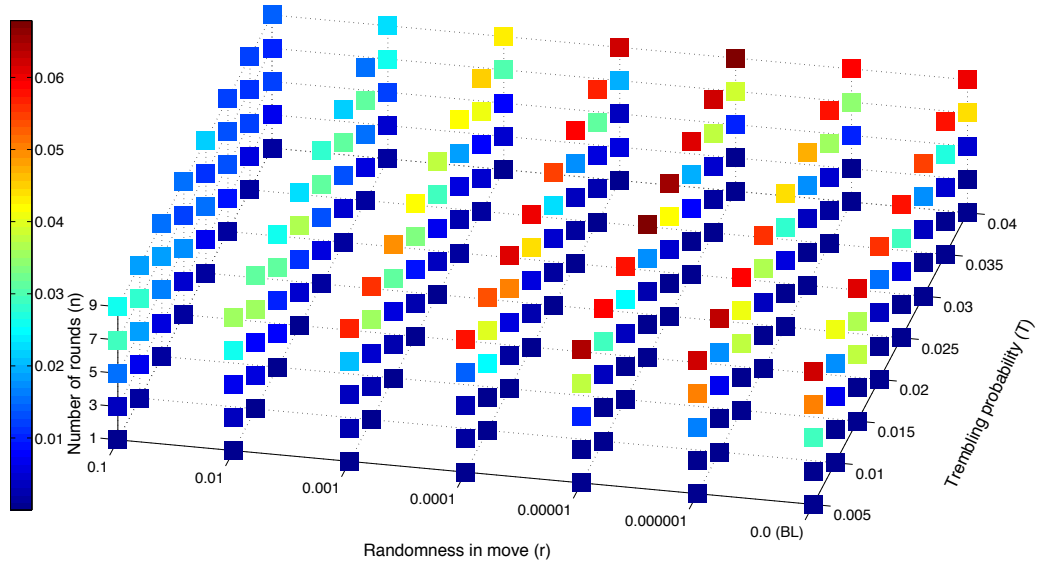


Figure B.91: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

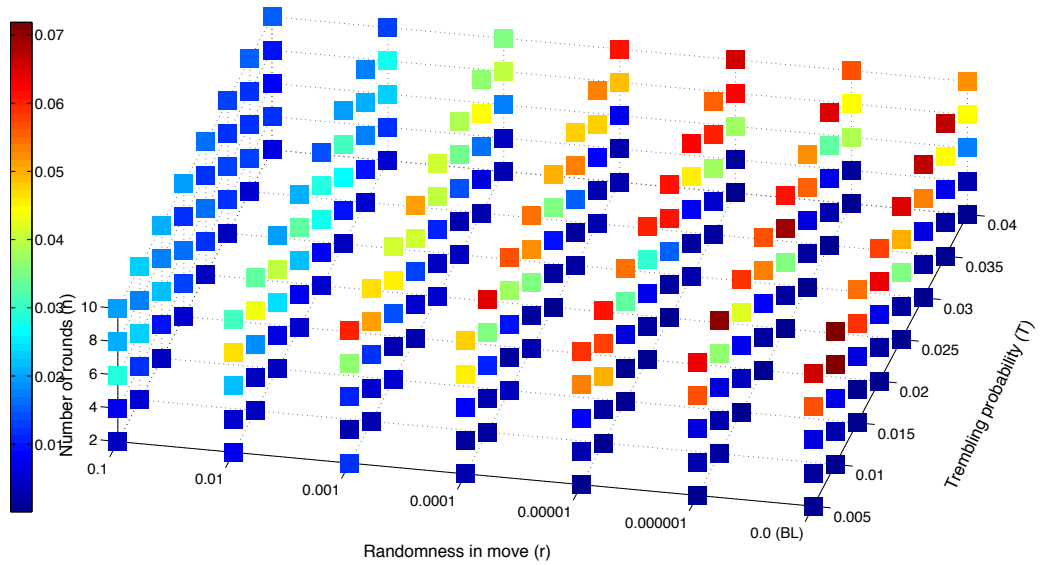


Figure B.92: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

odd and even respectively, for perception errors.

Error at 95% confidence for average maximum payoff

Figures B.95 and B.96 show the *error at* 95% *confidence for average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures B.97 and B.98 show the *error at* 95% *confidence for average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Error at 95% confidence for average minimum payoff

Figures B.99 and B.100 show the *error at* 95% *confidence for average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

Figures B.101 and B.102 show the *error at* 95% *confidence for average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for perception errors.

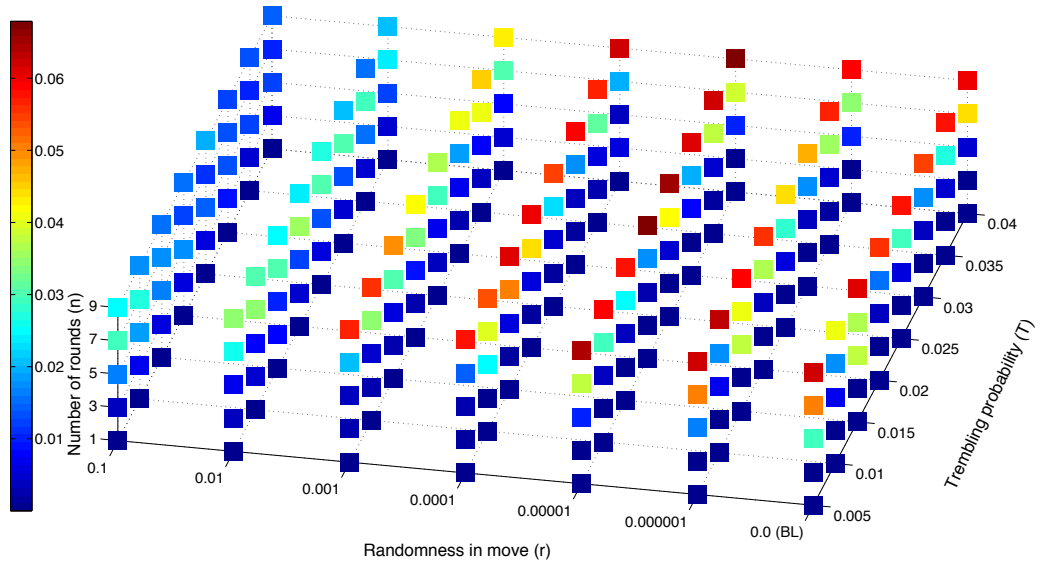


Figure B.93: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

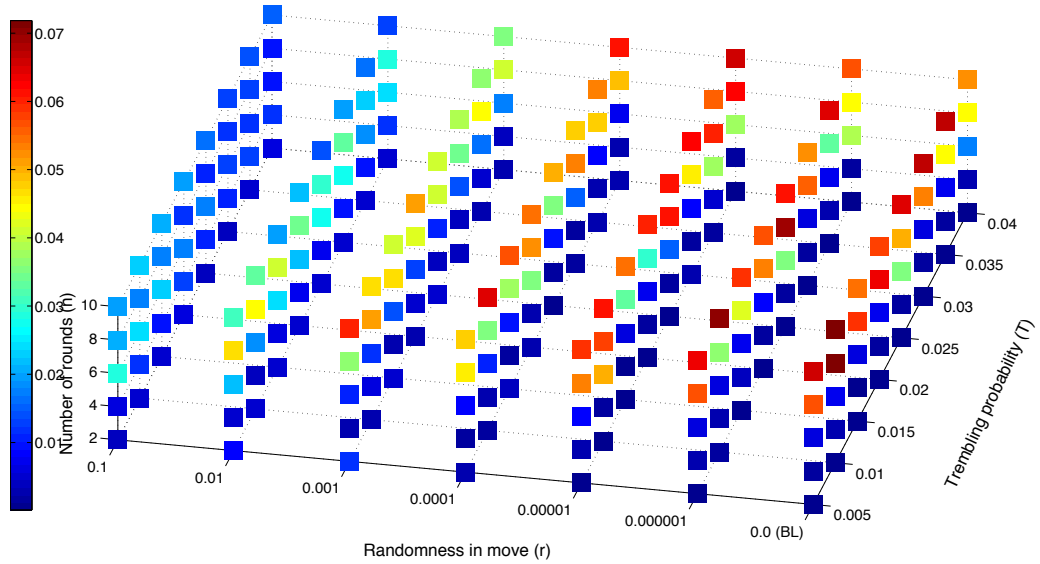


Figure B.94: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

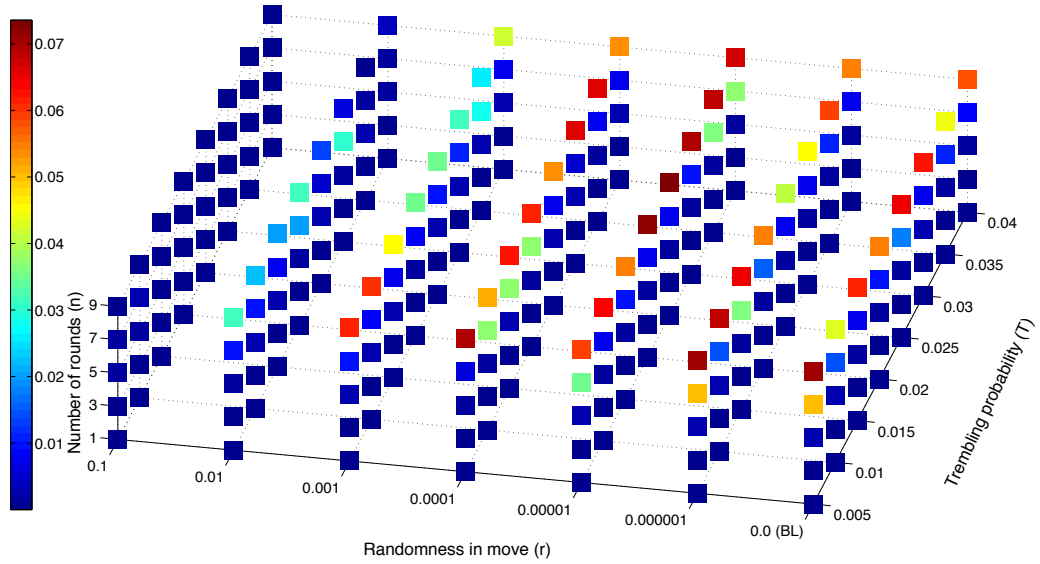


Figure B.95: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

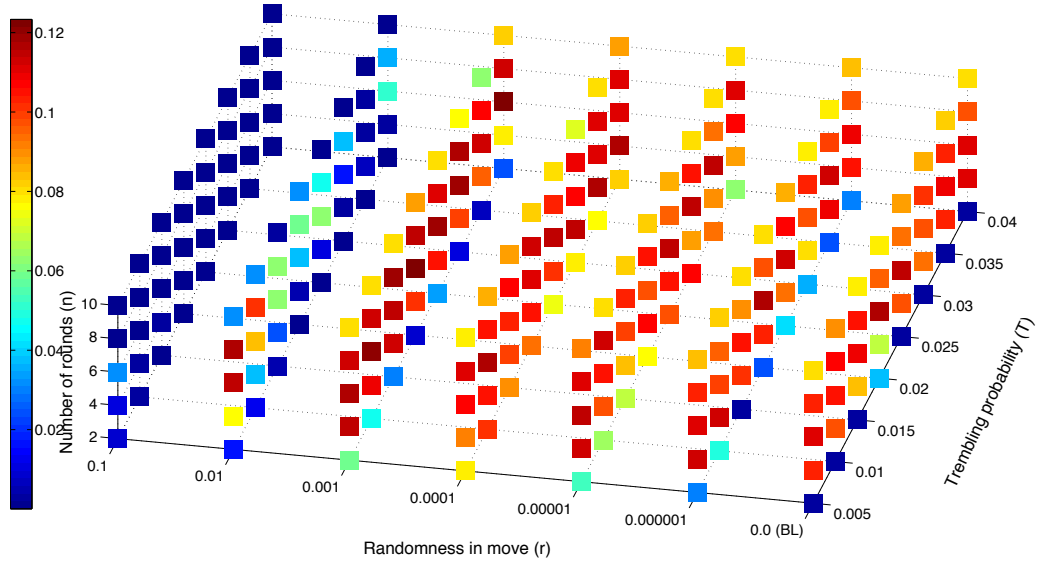


Figure B.96: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

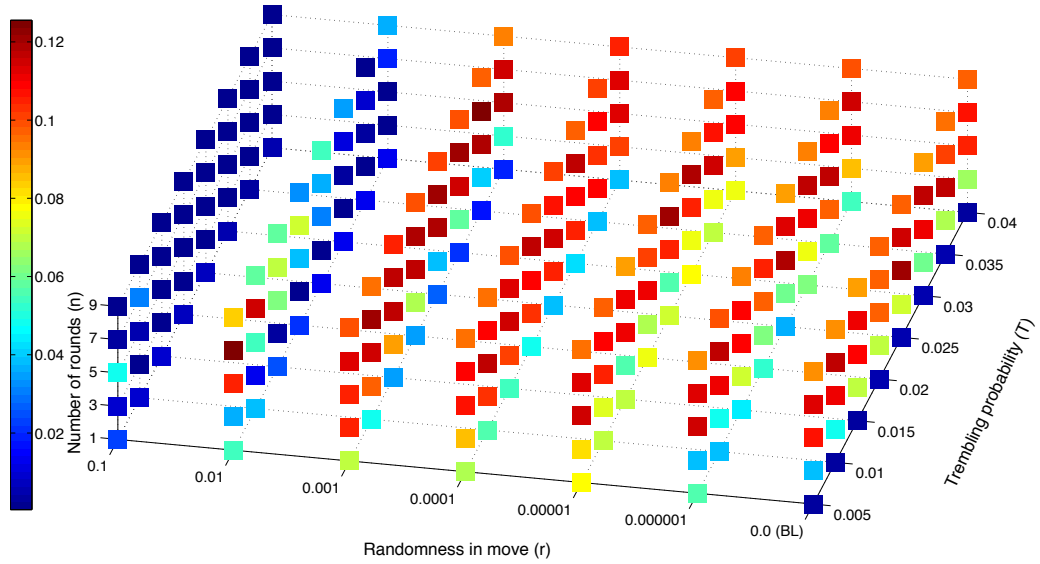


Figure B.97: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

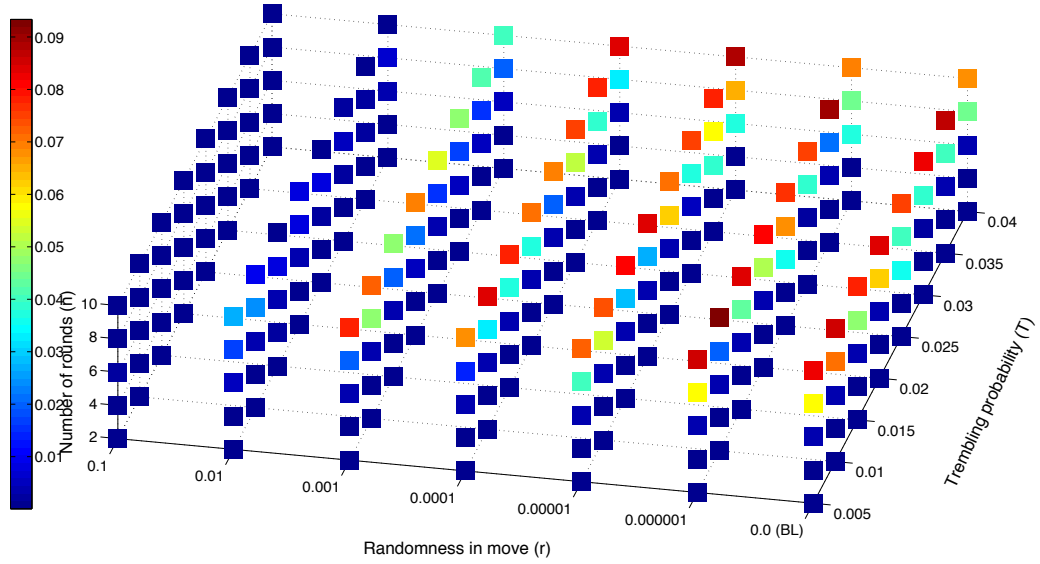


Figure B.98: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

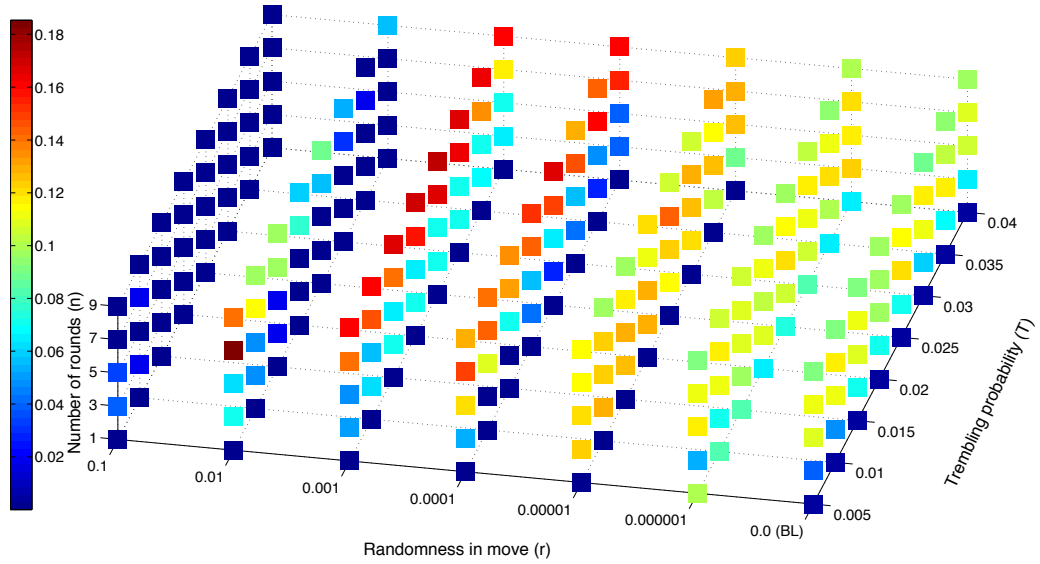


Figure B.99: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

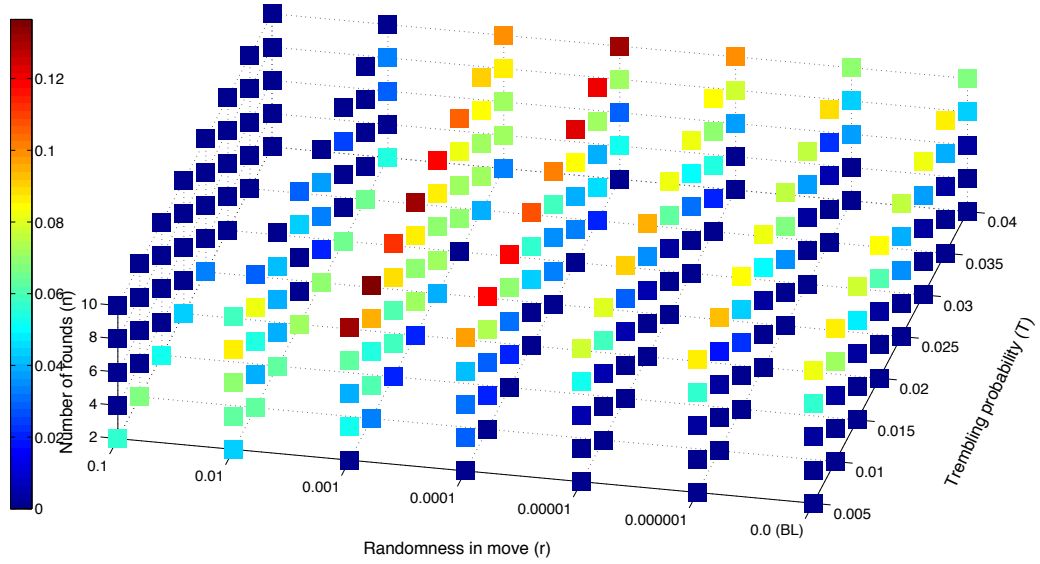


Figure B.100: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

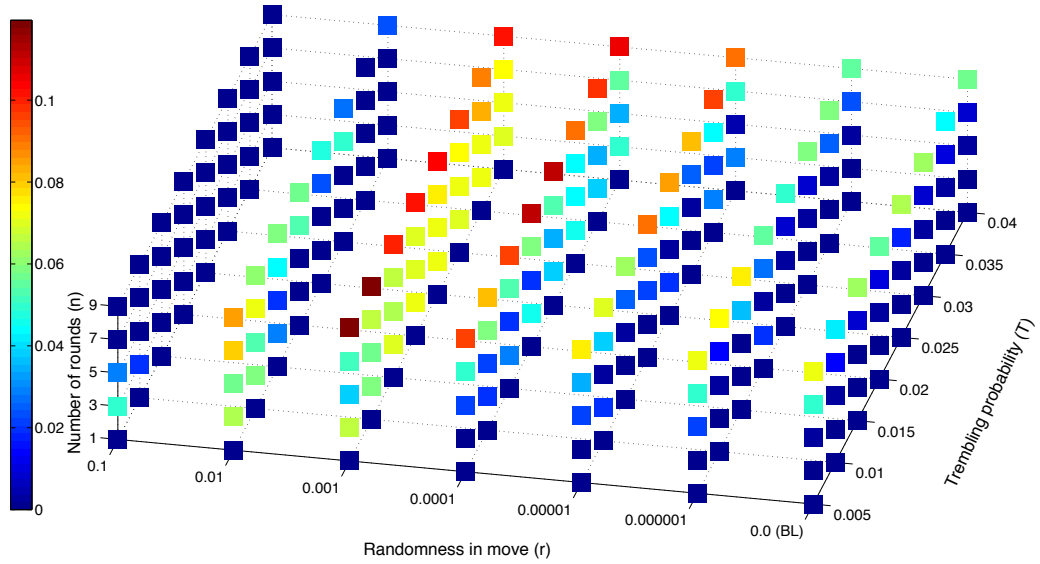


Figure B.101: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (perception errors).

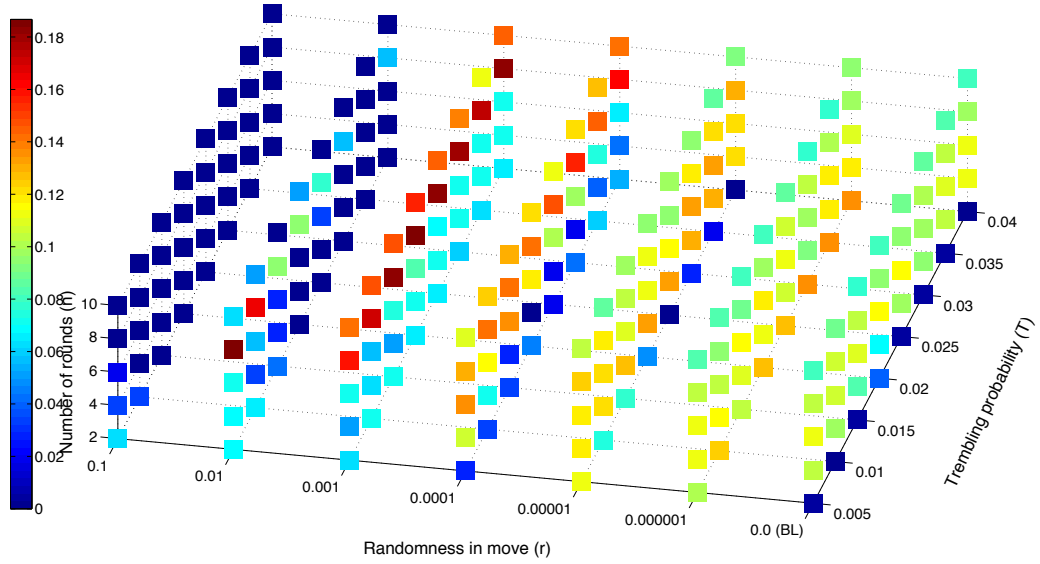


Figure B.102: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (perception errors).

B.3.3 Implementation and Perception Errors

Error at 95% confidence for average first hitting time

Figure B.103 shows the *error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n, for implementation and perception errors.

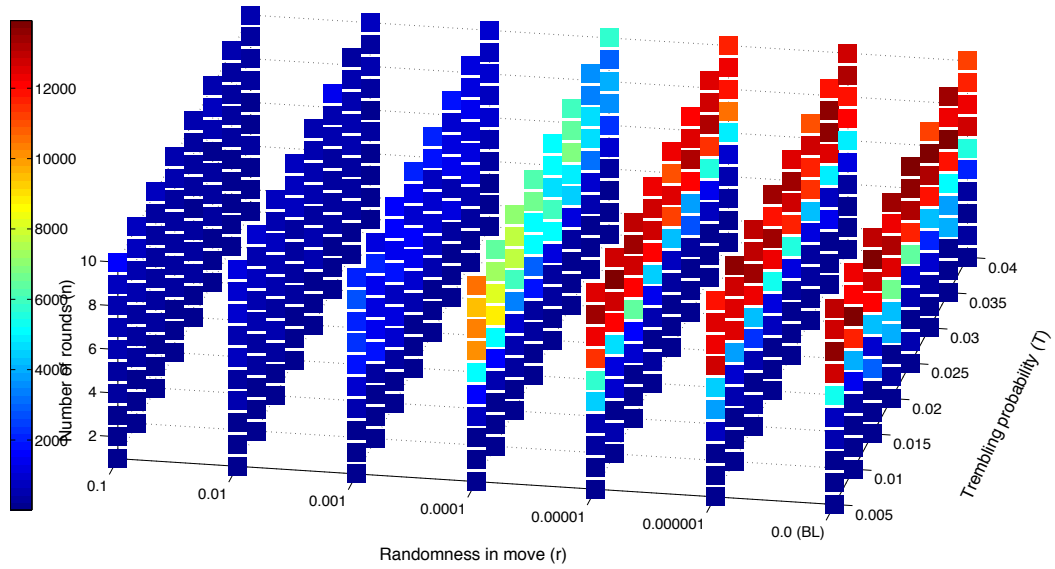


Figure B.103: *Error at* 95% *confidence for average first hitting time*, across 50 runs for all values of r, T and n (implementation and perception errors).

Error at 95% confidence for average stay in equilibrium

Figure B.104 shows the *error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n, for implementation and perception errors.

Error at 95% confidence for average distance from equilibrium

Figure B.105 shows the *error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n, for implementation and perception errors.

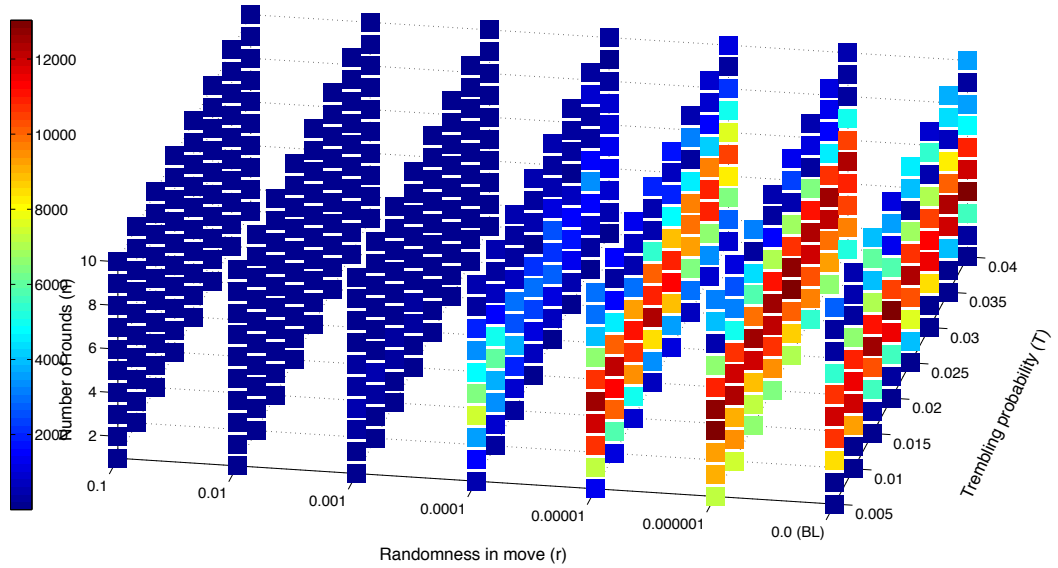


Figure B.104: *Error at* 95% *confidence for average stay within* ϵ *distance from equilibrium*, across 50 runs for all values of r, T and n (implementation and perception errors).

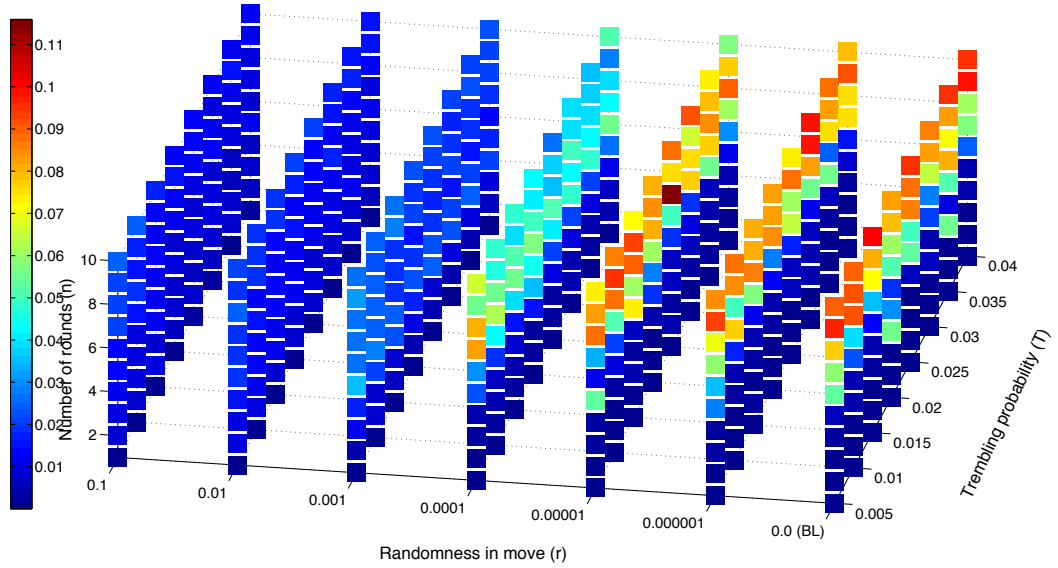


Figure B.105: *Error at* 95% *confidence for average distance from equilibrium, after first hit*, across 50 runs for all values of r, T and n (implementation and perception errors).

Error at 95% confidence for average payoff

Figures B.106 and B.107 show the *error at* 95% *confidence for average payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

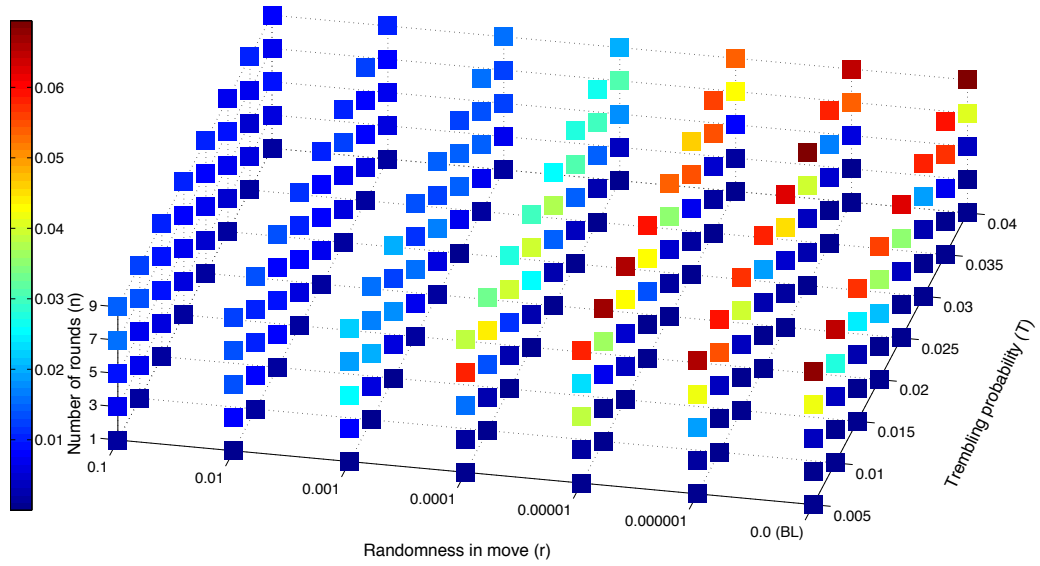


Figure B.106: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

Figures B.108 and B.109 show the *error at* 95% *confidence for average payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Error at 95% confidence for average maximum payoff

Figures B.110 and B.111 show the *error at* 95% *confidence for average maximum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Figures B.112 and B.113 show the *error at* 95% *confidence for average maximum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

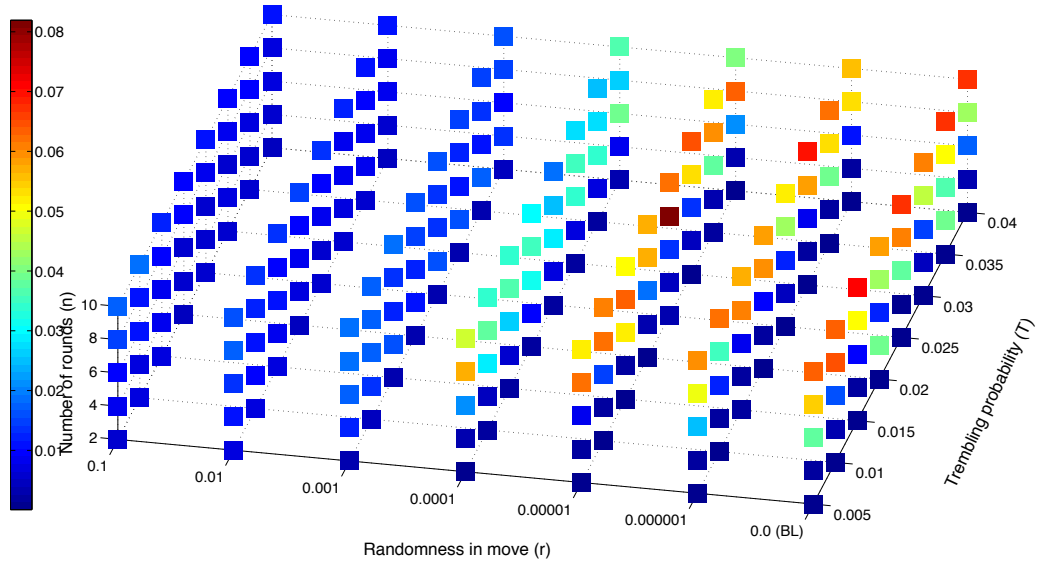


Figure B.107: *Error at* 95% *confidence for average payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

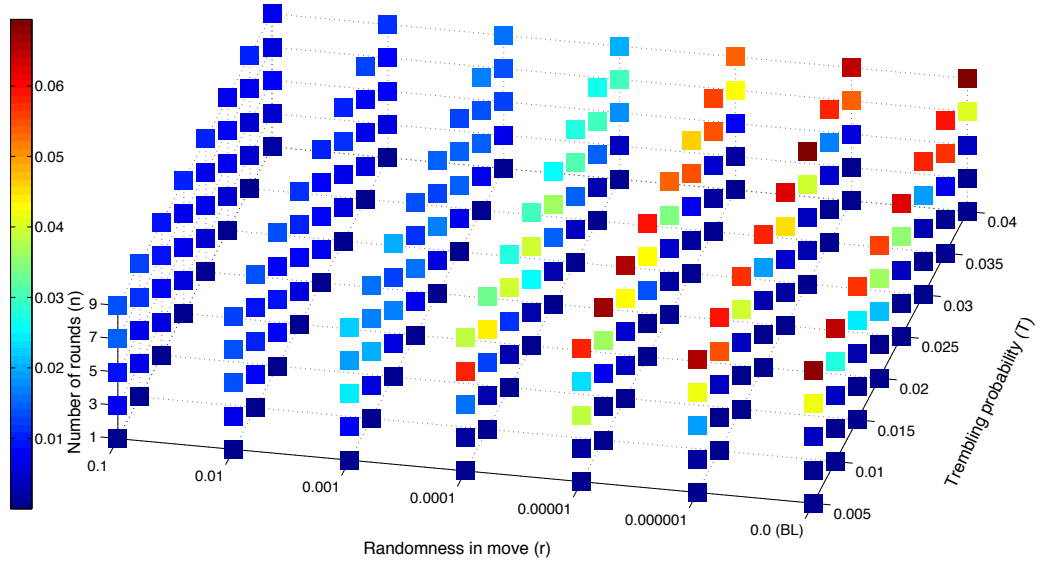


Figure B.108: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

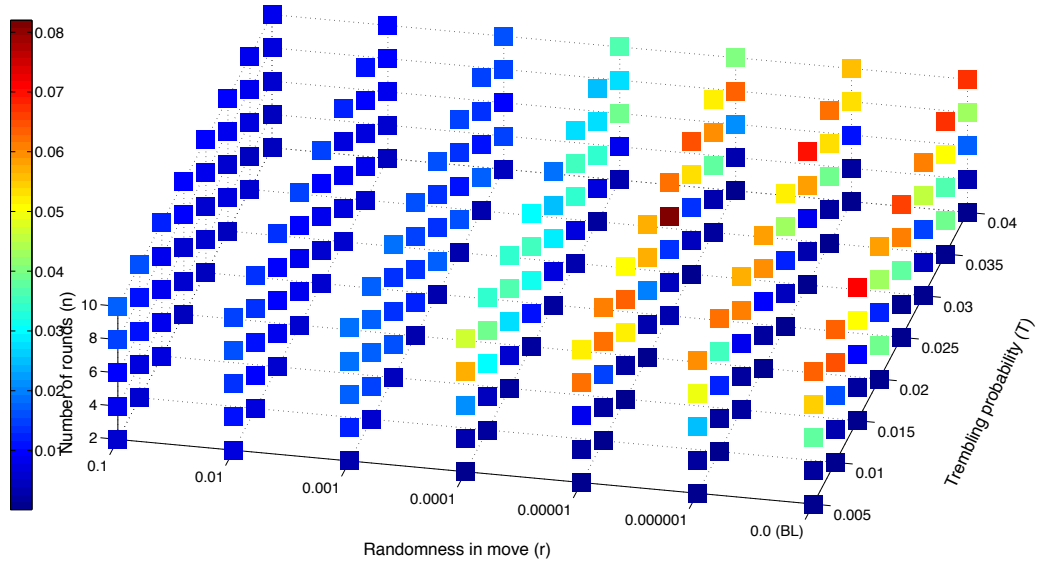


Figure B.109: *Error at* 95% *confidence for average payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

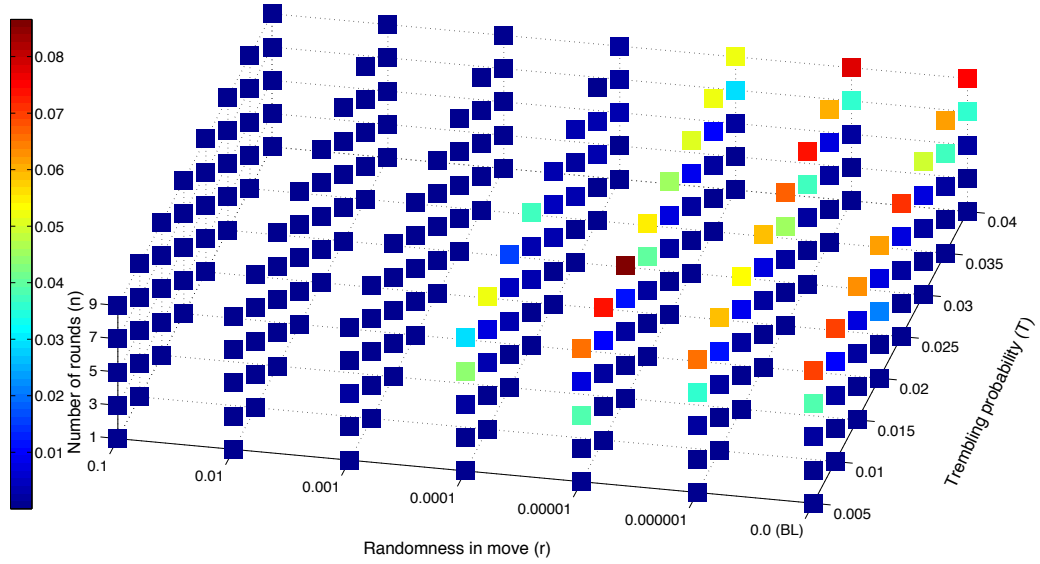


Figure B.110: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

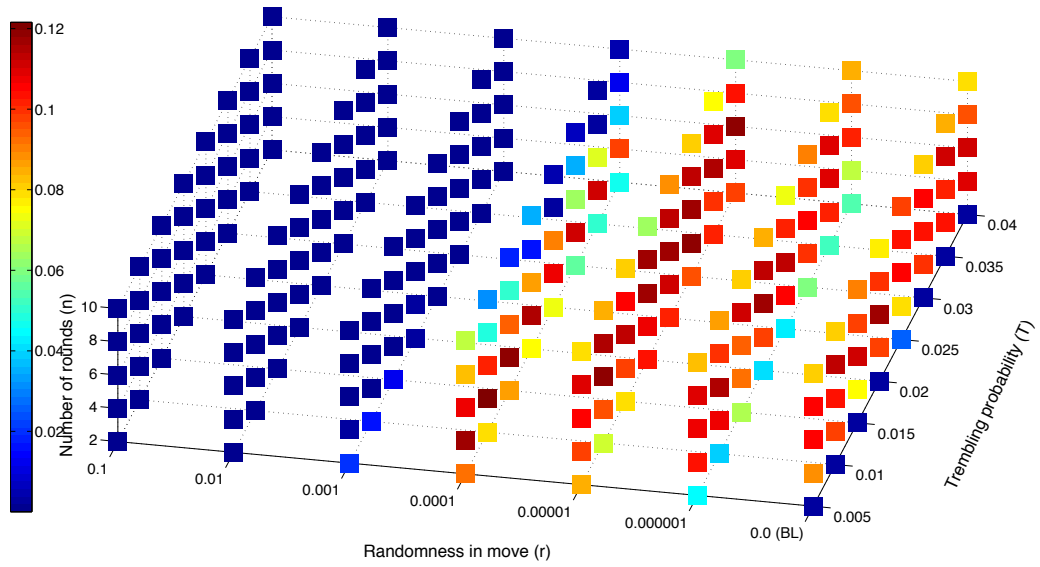


Figure B.111: *Error at* 95% *confidence for average maximum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

Error at 95% confidence for average minimum payoff

Figures B.114 and B.115 show the *error at* 95% *confidence for average minimum payoff for Player 1, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

Figures B.116 and B.117 show the *error at* 95% *confidence for average minimum payoff for Player 2, after first hit*, across 50 runs for all values of r, T and for n being odd and even respectively, for implementation and perception errors.

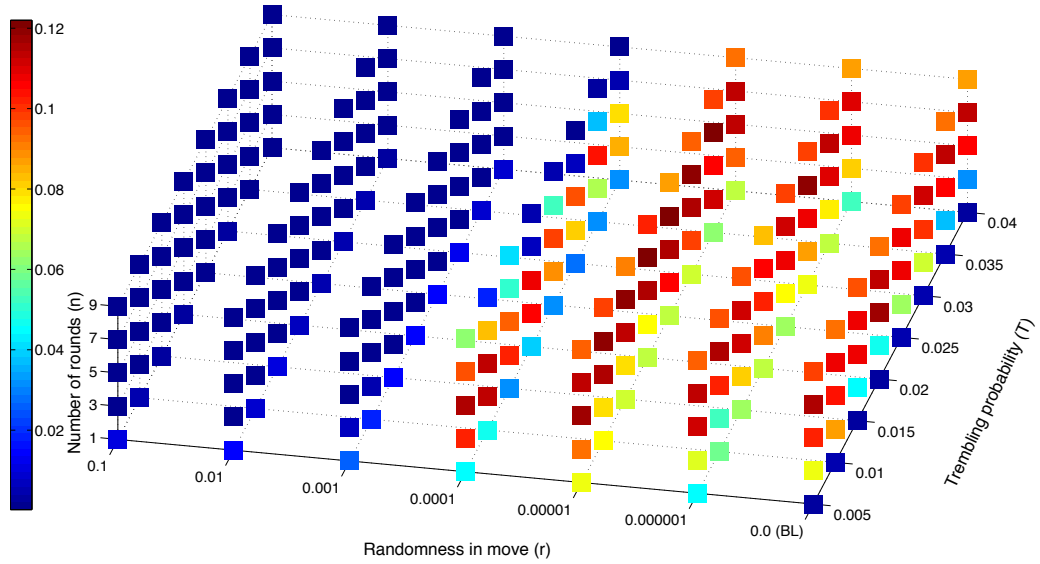


Figure B.112: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

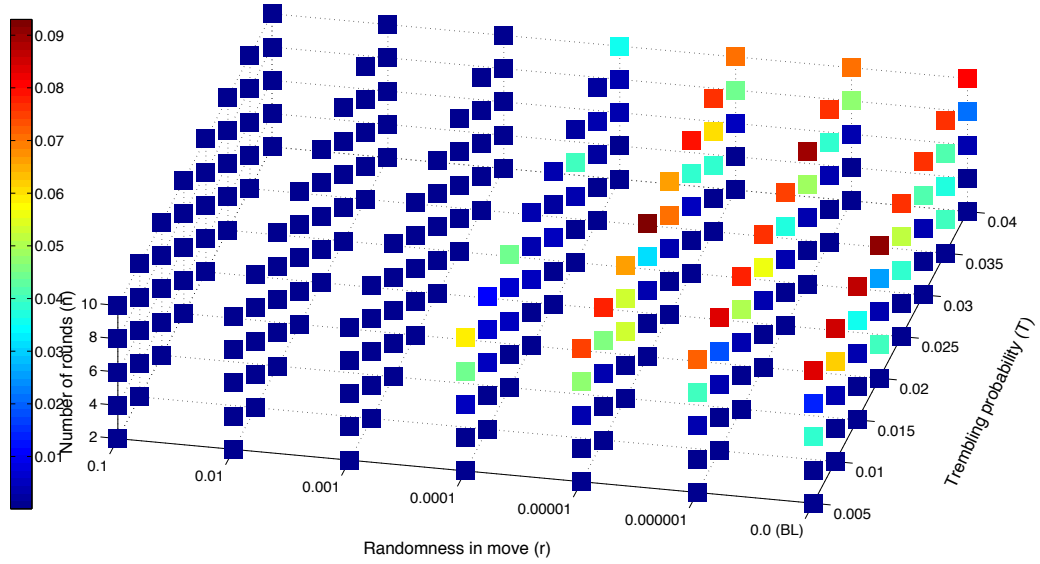


Figure B.113: *Error at* 95% *confidence for average maximum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

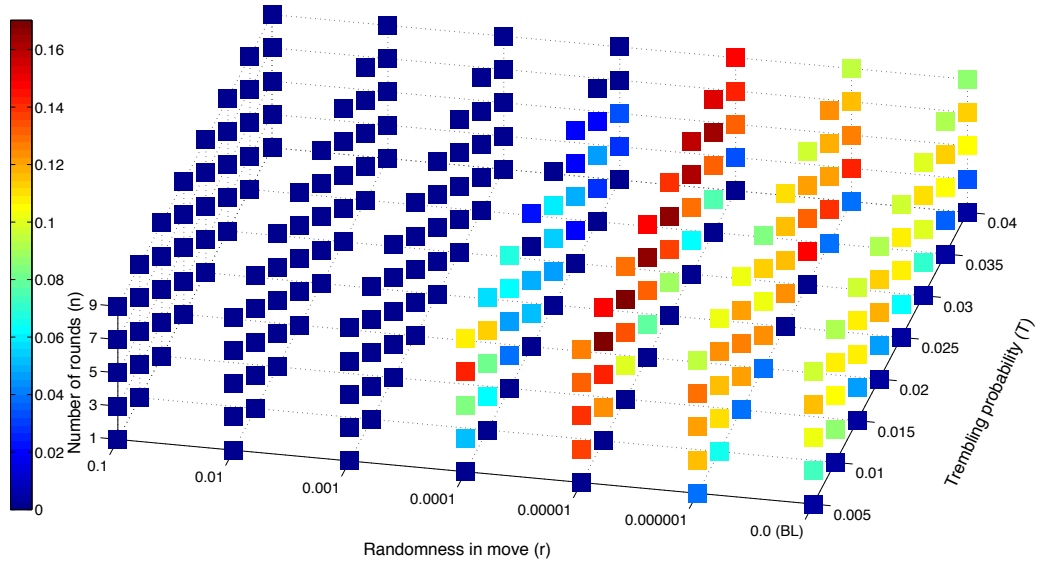


Figure B.114: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

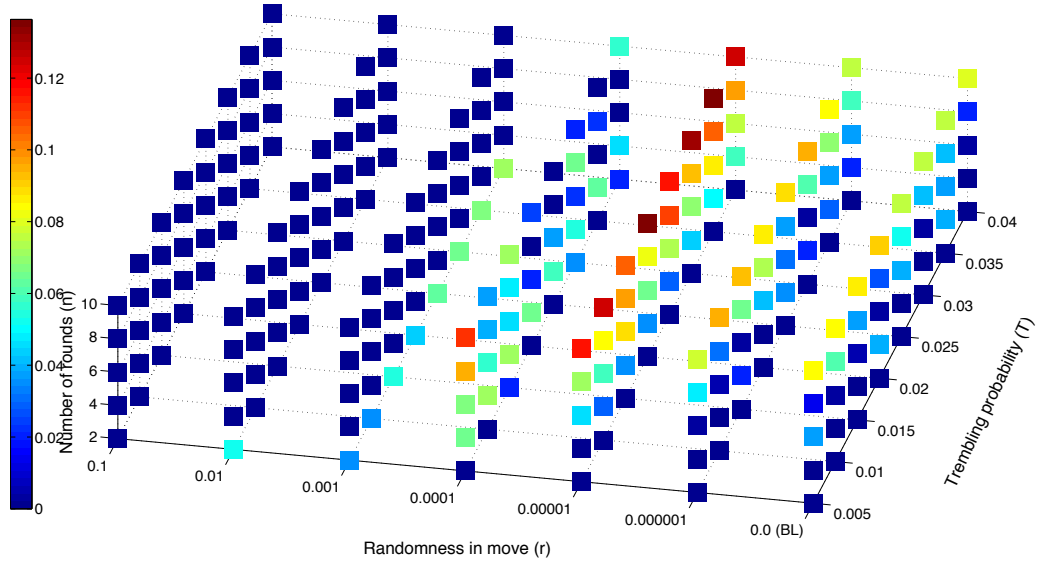


Figure B.115: *Error at* 95% *confidence for average minimum payoff (Player 1), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

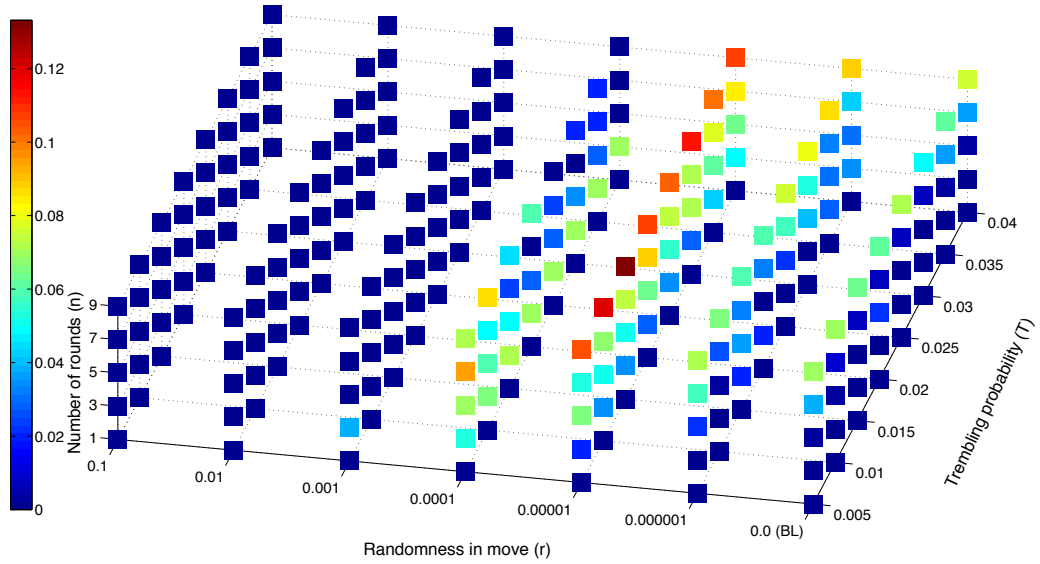


Figure B.116: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being odd (implementation and perception errors).

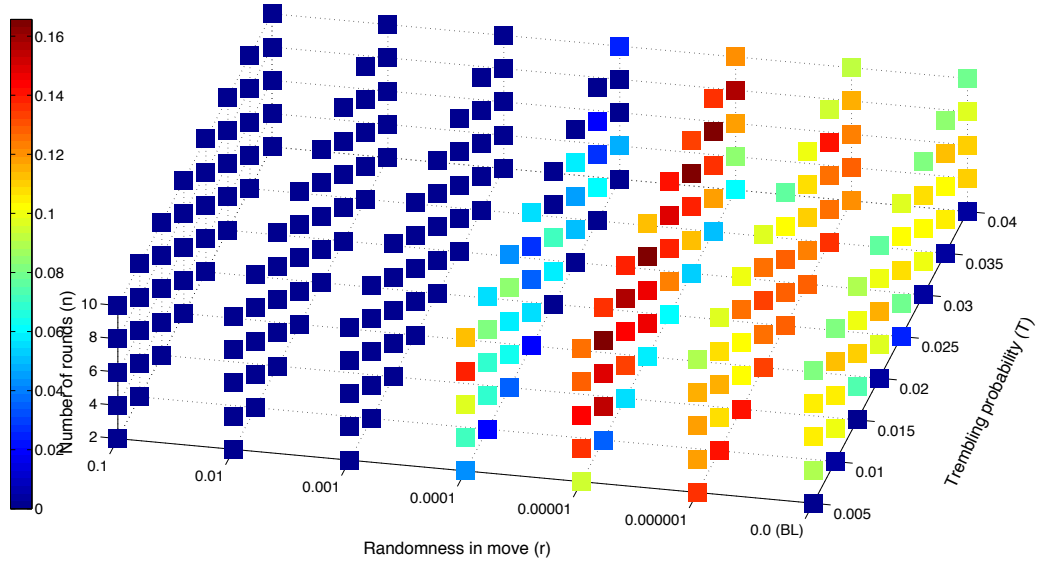


Figure B.117: *Error at* 95% *confidence for average minimum payoff (Player 2), after first hit*, across 50 runs for all values of r, T and for n being even (implementation and perception errors).

References

[1] F. Alkemade, *Evolutionary agent-based economics*, Ph.D. thesis, Technische Universiteit Eindhoven, 2004. 3, 4, 7, 51, 52, 54, 64, 74, 75, 169

[2] F. Alkemade, J.A. La Poutré, and H.M. Amman, *Robust evolutionary algorithm design for socio-economic simulation*, Computational Economics **28** (2006), no. 4, 355–370. 4, 54, 59, 60, 62, 76, 172, 173

[3] M.R. Andersson and T.W. Sandholm, *Leveled commitment contracts with myopic and strategic agents*, Journal of Economic Dynamics and Control **25** (2001), no. 3-4, 615–640. 52

[4] J. Andreoni and J.H. Miller, *Auctions with artificial adaptive agents*, Games and Economic Behavior **10** (1995), no. 1, 39–64. 51, 61

[5] J. Arifovic, *Genetic algorithm learning and the cobweb model*, Journal of Economic dynamics and Control **18** (1994), no. 1, 3–28. 3, 51, 52, 61, 63, 173

[6] ———, *The behavior of the exchange rate in the genetic algorithm and experimental economies*, Journal of Political Economy (1996), 510–541. 51

[7] J. Arifovic and C. Eaton, *Coordination via genetic learning*, Computational Economics **8** (1995), no. 3, 181–203. 51

[8] J. Arifovic and M.K. Maschek, *Revisiting individual evolutionary learning in the cobweb model – an illustration of the virtual spite-effect*, Computational economics **28** (2006), no. 4, 333–354. 4, 7, 62, 76, 173

[9] W.B. Arthur, *On designing economic agents that behave like human agents*, Journal of Evolutionary Economics **3** (1993), no. 1, 1–22. 51

[10] W.B. Arthur, J.H. Holland, B. LeBaron, R. Palmer, and P. Tayler, *Asset pricing under endogenous expectations in an artificial stock market*, The economy as an evolving complex system II: Proceedings of SFI Studies in the Sciences of Complexity, Addison-Wesley, 1997, pp. 15–44. 50

[11] R.M. Axelrod, *Genetic algorithms and simulated annealing*, ch. The evolution of strategies in the iderated prisoner's dilemma, pp. 32–41, Morgan Kaufmann, 1987. 59

[12] ———, *The complexity of cooperation: Agent-based models of competition and collaboration*, Princeton University Press, 1997. 3, 51

[13] R.L. Axtell, J.M. Epstein, and H.P. Young, *The emergence of classes in a multiagent bargaining model*, Social Dynamics (1999), 191–211. 51

[14] M. Baca and H. Raff, *Issue-by-issue negotiations: The role of information and time preference*, Games and Economic Behavior **13** (1996), no. 1, 125–134. 35

[15] O.J. Bartos, *Modeling distributive and integrative negotiations*, Annals of the American Academy of Political and Social Science **542** (1995), 48–60. 28

[16] D.E. Bell, *Regret in decision making under uncertainty*, Operations Research **30** (1982), no. 5, 961–981. 57

[17] H-G. Beyer, *The theory of evolution strategies*, Springer-Verlag, New York, 2001. 81, 88, 322

[18] K. Binmore, *Essays on the foundations of game theory*, Basil Blackwell, 1990. 23

[19] ———, *Fun and games: A text on game theory*, D. C. Heath and Company, 1992. 26, 27, 31, 34

[20] K. Binmore, A. Rubinstein, and A. Wolinsky, *The Nash bargaining solution in economic modelling*, The RAND Journal of Economics **17** (1986), no. 2, 176–188. 34

[21] E. Bonabeau, *Agent-based modeling: Methods and techniques for simulating human systems*, Proceedings of the National Academy of Sciences of the United States of America **99** (2002), no. Suppl 3, 7280. 6, 7

[22] T. Brenner, *Agent learning representation: Advice on modelling economic learning*, Handbook of Computational Economics, vol. 2, Elsevier, 2006, pp. 895 – 947. 58

[23] S.G. Bullock, *Co-evolutionary design: Implications for evolutionary robotics*, The 3rd European Conference on Artificial Life, 1995. 42

[24] L-A. Busch and I.J. Horstmann, *Signaling via an agenda in multi-issue bargaining with incomplete information*, Economic Theory **13** (1999), no. 3, 561–575. 35

[25] N.T. Chan, B. LeBaron, A.W. Lo, and T. Poggio, *Agent-based models of financial markets: A comparison with experimental markets*, Working Paper No. 124, MIT Artificial Markets Project, MIT, MA (1999). 51

[26] A. Chandra, P.S. Oliveto, and X. Yao, *Co-evolution of optimal agents for the alternating offers bargaining game*, Proceedings of the European Conference on the Applications of Evolutionary Computation (EvoApplications 2010), Springer-Verlag, 2010, pp. 61–70. 18

[27] E. Chattoe, *Just how (un)realistic are evolutionary algorithms as representations of social processes?*, Journal of Artificial Societies and Social Simulation **1** (1998), no. 3. 47, 49, 51, 71

[28] S.Y. Chong, *Generalization and diversity in co-evolutionary learning*, Ph.D. thesis, University of Birmingham, 2007. 39, 59, 66

[29] S.Y. Chong and X. Yao, *Behavioral diversity, choices, and noise in the iterated prisoners dilemma*, IEEE Transactions on Evolutionary Computation **9** (2005), 540–551. 51, 126

[30] S. H. Clearwater, *Market based control: A paradigm for distributed resource allocation*, World Scientific, 1996. 47, 164

[31] D. Cliff, *Zip60: Further explorations in the evolutionary design of online auction market mechanisms*, Tech. Report HPL-2005-85, Hewlett-Packard Laboratories, Jan 2005. 48, 53

[32] D. Cliff and J. Bruten, *Minimal-intelligence agents for bargaining behaviors in market-based environments*, Tech. Report HPL-97-91, Hewlett-Packard Laboratories, 1997. 48, 53

[33] D. Cliff and G. Miller, *Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations*, Advances In Artificial Life (1995), 200–218. 42, 76

[34] R.M. Coehoorn and N.R. Jennings, *Learning on opponent's preferences to make effective multi-issue negotiation trade-offs*, Proceedings of the 6th International Conference on Electronic Commerce, ACM, 2004, pp. 59–68. 37, 53

[35] A. Cournot, *Researches into the mathematical principles of the theory of wealth*, Macmillan, New York, 1897. 59, 63

[36] P.J. Darwen, *Co-evolutionary learning by automatic modularisation with speciation*, Ph.D. thesis, University of New South Wales, 1996. 39, 42, 66

[37] P.J. Darwen and X. Yao, *Does extra genetic diversity maintain escalation in a co-evolutionary arms race*, International Journal of Knowledge-Based Intelligent Engineering Systems **4** (2000), no. 3, 191–200. 42

[38] ———, *Co-evolution in iterated prisoner's dilemma with intermediate levels of cooperation: Application to missile defense*, International Journal of Computational Intelligence and Applications **2** (2002), 83–108. 51

[39] H. Dawid, *Adaptive learning by genetic algorithms: Analytical results and applications to economic models*, Springer-Verlag, 1996. 3, 4, 41, 47, 48, 49, 51, 52, 59, 60, 61, 62, 64, 68, 71, 73, 74, 76, 119, 173, 174

[40] H. Dawid and J. Dermietzel, *How robust is the equal split norm? responsive strategies, selection mechanisms and the need for economic interpretation of simulation parameters*, Computational Economics **28** (2006), no. 4, 371–397. 86

[41] H. Dawid, M. Reimann, and B. Bullnheimer, *To innovate or not to innovate?*, IEEE Transactions on Evolutionary Computation **5** (2001), no. 5, 471–481. 53

[42] K.A. De Jong, W.M. Spears, and D.F. Gordon, *Using genetic algorithms for concept learning*, Machine Learning **13** (1993), no. 2, 161–188. 4, 49, 72

[43] C. Dibble and P.G. Feldman, *The GeoGraph 3d computational laboratory: network and terrain landscapes for repast*, Journal of Artificial Societies and Social Simulation **7** (2004), no. 1. 47

[44] J. Duffy, *Learning to speculate: Experiments with artificial and real agents*, Journal of Economic Dynamics and Control **25** (2001), no. 3-4, 295–319. 52, 62

[45] F.Y. Edgeworth, *Mathematical psychics: An essay on the application of mathematics to moral sciences*, Kegan Paul Publishers, 1881. 22

[46] B. Edmonds, *Towards a descriptive model of agent strategy search*, Computational Economics **18** (2001), no. 1, 111–133. 49, 51

[47] T.S. Ellis and X. Yao, *Evolving cooperation in the non-iterated prisoners dilemma: A social network inspired approach*, IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 736–743. 51

[48] J.M. Epstein, *Learning to be thoughtless: Social norms and individual computation*, Computational Economics **18** (2001), no. 1, 9–24. 51

[49] J.M. Epstein and R. Axtell, *Growing artificial societies*, MIT press Cambridge, MA, 1996. 51

[50] I. Erev and A.E. Roth, *Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria*, American Economic Review **88** (1998), no. 4, 848–881. 47, 71

[51] C. Fershtman, *The importance of the agenda in bargaining*, Games and Economic Behavior **2** (1990), no. 3, 224–238. 35

[52] ———, *A note on multi-issue two-sided bargaining: Bilateral procedures*, Games and Economic Behavior **30** (2000), no. 2, 216–227. 35

[53] S.G. Ficici, *Solution concepts in coevolutionary algorithms*, Ph.D. thesis, Brandeis University, 2004. 4, 5, 8, 41, 42, 43, 44, 45, 66, 67, 76, 86

[54] S.G. Ficici and J. Pollack, *A game-theoretic approach to the simple coevolutionary algorithm*, Parallel Problem Solving from Nature PPSN VI, Springer, 2000, pp. 467–476. 41

[55] M.M. Flood, *Some experimental games*, Management Science **5** (1958), no. 1, 5. 56

[56] B.S. Frey, *"Just forget it." Memory distortions as bounded rationality*, Mind & Society **4** (2005), no. 1, 13–25. 121, 126

[57] D. Fudenberg and D.K. Levine, *The theory of learning in games*, The MIT press, 1998. 47

[58] E.H. Gerding, *Autonomous agents in bargaining games: An evolutionary investigation of fundamentals, strategies, and business applications*, Ph.D. thesis, Technische Universiteit Eindhoven, 2004. 4, 9, 23, 24, 28, 32, 36, 47, 50, 167

[59] E.H. Gerding, D.D.B. van Bragt, and J.A. La Poutré, *Multi-issue negotiation processes by evolutionary simulation, validation and social extensions*, Computational Economics **22** (2003), no. 1, 39–63. xi, 3, 11, 13, 16, 36, 47, 54, 69, 70, 74, 75, 77, 78, 79, 80, 81, 82, 85, 87, 88, 89, 90, 91, 107, 113, 117, 120, 150

[60] G. Gigerenzer and R. Selten, *Bounded rationality: The adaptive toolbox*, the MIT Press, 2002. 24, 56

[61] I. Gilboa and D. Schmeidler, *Case-based decision theory*, The Quarterly Journal of Economics **110** (1995), no. 3, 605. 57

[62] H. Gintis, *Game theory evolving*, Princeton University Press, 2000. 53, 71

[63] D.K. Gode and S. Sunder, *Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality*, Journal of Political Economy **101** (1993), no. 1, 119–137. 48, 53

[64] D.E. Goldberg, *Genetic algorithms in search, optimisation, and machine learning*, Addison-Wesley, 1989. 323

[65] ———, *The existential pleasures of genetic algorithms*, Genetic Algorithms in Engineering and Computer Science, Wiley and Sons, 1994, pp. 23–31. 54, 71

[66] M. Gsell, *Assessing the impact of algorithmic trading on markets: A simulation approach*, SSRN eLibrary. 52

[67] W. Güth, R. Schmittberger, and B. Schwarze, *An experimental analysis of ultimatum bargaining*, Journal of Economic Behavior & Organization **3** (1982), no. 4, 367–388. 57

[68] R.H. Guttman and P. Maes, *Cooperative vs. competitive multi-agent negotiations in retail electronic commerce*, Proceedings of the 2nd International Workshop on Cooperative Information Agents (CIA '98), Springer-Verlag, 1998, pp. 135–147. 28

[69] J.C. Harsanyi, *Approaches to the bargaining problem before and after the theory of games: A critical discussion of Zeuthen's, Hicks' and Nash's theories*, Econometrica **24** (1956), no. 2, 144–157. 22

[70] ———, *Games with incomplete information played by bayesian players, parts i-iii*, Management Science **14** (1967–1968), 159–182, 320–334, 486–502. 23, 37

[71] F.A. Hayek, *Individualism and economic order*, University of Chicago Press, 1948. 46

[72] W.D. Hillis, *Co-evolving parasites improve simulated evolution as an optimization procedure*, Pysica D: Nonlinear Phenomena **42** (1990), no. 1–3, 228–234. 38, 42, 66

[73] K. Hindriks and D. Tykhonov, *Opponent modelling in automated multi-issue negotiation using bayesian learning*, Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, 2008, pp. 331–338. 53

[74] J.H. Holland, *Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems*, Computation & Intelligence, AAAI, 1995, pp. 275–304. 3, 49, 72, 324

[75] ———, *Hidden order: How adaptation builds complexity*, Basic Books, 1996. 51

[76] G.S. Hornby and B. Mirtich, *Comparing diffuse and true coevolution in a physics-based world*, Tech. Report TR-98-11, Mitsubishi Electric Research Laboratories, 1999. 42

[77] Y. In and R. Serrano, *Agenda restrictions in multi-issue bargaining (ii): unrestricted agendas*, Economics Letters **79** (2003), no. 3, 325–331. 35

[78] R. Inderst, *Multi-issue bargaining with endogenous agenda*, Games and Economic Behavior **3** (2000), no. 1, 64–82. 35

[79] C.A. Ioannou, *Algorithmic bounded rationality, optimality and noise*, Ph.D. thesis, University of Minnesota, 2009. 15, 47, 56, 121, 124, 125, 127, 163

[80] N. Jin, *Constraint-based co-evolutionary genetic programming for bargaining problems*, Ph.D. thesis, University of Essex, 2007. 66

[81] H. Juille and J.B. Pollack, *Dynamics of co-evolutionary learning*, Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, MIT Press, 1996, pp. 526–534. 42

[82] D. Kahneman and A. Tversky, *Prospect theory: An analysis of decision under risk*, Econometrica **47** (1979), 263–291. 57

[83] E. Kalai and M. Smorodinsky, *Other solutions to Nash's bargaining problem*, Econometrica **43** (1975), no. 3, 513–518. 27

[84] J.O. Kephart, *Software agents and the route to the information economy*, Proceedings of the National Academy of Sciences **99** (2002), no. Suppl 3, 7207–7213. 53, 73

[85] A.P. Kirman and N.J. Vriend, *Evolving market structure: An ACE model of price dispersion and loyalty*, Journal of Economic Dynamics and Control **25** (2001), no. 3-4, 459–502. 51

[86] T.B. Klos, *Decentralized interaction and co-adaptation in the repeated prisoner's dilemma*, Computational & Mathematical Organization Theory **5** (1999), no. 2, 147–165. 49, 51

[87] S. Kraus, *Beliefs, time and incomplete information in multiple encounter negotiations among autonomous agents*, Annals of Mathematics and Artificial Intelligence **20** (1997), 111–159. 37

[88] S. Lahiri, *Axiomatic characterization of the Nash and Kalai-Smorodinsky solutions for discrete bargaining problems*, Pure Mathematics and Applications **14** (2003), no. 3, 207–220. 26

[89] B. LeBaron, *Empirical regularities from interacting long- and short-memory investors in an agent-based stock market*, IEEE Transactions on Evolutionary Computation **5** (2001), no. 5, 442–455. 50

[90] H. Lipson and J.B. Pollack, *Automatic design and manufacture of robotic lifeforms*, Nature **406** (2000), no. 6799, 974–978. 39, 66

[91] R. Litterman and P. Knez, *Genetic algorithm for the Kiyotaki-Wright model*, Tech. report, Mimeo, Goldman-Sachs, 1989. 61

[92] R.D. Luce and H. Raiffa, *Games and decisions*, Wiley, New York, 1957. 27

[93] R. Marimon, E. McGrattan, and T.J. Sargent, *Money as a medium of exchange in an economy with artificially intelligent agents*, Journal of Economic Dynamics and Control **14** (1990), 329–73. 47, 61, 71

[94] R.E. Marks, *Breeding hybrid strategies: Optimal behaviour for oligopolists*, Journal of Evolutionary Economics **2** (1992), no. 1, 17–38. 3, 50, 72

[95] ———, *Evolved perception and behaviour in oligopolies*, Journal of Economic Dynamics and Control **22** (1998), no. 8-9, 1209–1233. 49

[96] ———, *Validating simulation models: a general framework and four applied examples*, Computational Economics **30** (2007), no. 3, 265–290. 4, 7, 52

[97] N. Matos, C. Sierra, and N. R. Jennings, *Determining successful negotiation strategies: An evolutionary approach*, Proc. of the 3rd International Conference on Multi-Agent Systems (ICMAS-98), 1998, pp. 182–189. 36

[98] T. Miconi, *The road to everywhere: Evolution, complexity and progress in natural and artificial systems*, Ph.D. thesis, University of Birmingham, 2007. 38, 39, 41, 42, 45

[99] J.H. Miller and J. Andreoni, *Can evolutionary dynamics explain free riding in experiments*, Economics Letters **36** (1991), no. 1, 9–15. 51

[100] R.B. Myerson, *Utilitarianism, egalitarianism, and the timing effect in social choice problems*, Econometrica **49** (1981), no. 4, 883–897. 27, 28

[101] J. Nash, *The bargaining problem*, Econometrica **18** (1950), no. 2, 155–162. 25, 26

[102] ———, *Equilibrium points in n-person games*, Proceedings of the National Academy of Sciences **36** (1950), no. 1, 48–49. 29

[103] ———, *Non-cooperative games*, The Annals of Mathematics **54** (1951), no. 2, 286–295. 25, 29

[104] ———, *Two-person cooperative games*, Econometrica **21** (1953), no. 1, 128–140. 25, 30

[105] A.E. Nix and M.D. Vose, *Modeling genetic algorithms with Markov chains*, Annals of Mathematics and Artificial Intelligence **5** (1992), no. 1, 79–88. 62, 173

[106] S. Nolfi and D. Floreano, *Coevolving predator and prey robots: Do "arms races arise in artificial evolution?*, Artificial Life **4** (1998), no. 4, 311–335. 5, 38, 41, 42, 66

[107] J.R. Oliver, *A machine learning approach to automated negotiation and prospects for electronic commerce*, Journal of Management Information Systems **13** (1996), no. 3, 83–112. 36, 53

[108] M. Olson, *The logic of collective action: Public goods and the theory of groups*, Harvard University Press, 1971. 46

[109] B. Olsson, *NK-landscapes as test functions for evaluation of host-parasite algorithms*, Parallel Problem Solving from Nature (PPSN VI), Springer, 2000, pp. 487–496. 42

[110] M.J. Osborne and A. Rubinstein, *Bargaining and markets*, Academic Press, 1990. 34

[111] L. Pagie and P. Hogeweg, *Information integration and red queen dynamics in coevolutionary optimization*, Proceedings of the Congress on Evolutionary Computation (CEC 2000), IEEE Press, 2000, pp. 1260–1267. 42

[112] L. Pagie and M. Mitchell, *A comparison of evolutionary and coevolutionary search*, International Journal of Computational Intelligence and Applications **2** (2002), no. 1, 53–69. 41

[113] J. Paredis, *Towards balanced coevolution*, Parallel Problem Solving from Nature (PPSN VI), Springer, 2000, pp. 497–506. 42

[114] O.G. Paskelian, *The impact of algorithmic trading models on the stock market*, The Handbook of Trading: Strategies for Navigating and Profiting from Currency, Bond, and Stock Markets (2010), 275. 52

[115] S. Phelps, P. McBurney, S. Parsons, and E. Sklar, *Co-evolution of auction mechanisms and strategies: Towards a novel approach to microeconomic design*, Tech. Report ULCS-02-004, Computer Science Department, University of Liverpool, UK, 2002. 53

[116] S. Phelps, McBurney P., S. Parsons, and E. Sklar, *Co-evolutionary auction mechanism design*, Agent-Mediated Electronic Commerce IV (Berlin), Springer, 2002, pp. 123–142. 53

[117] M. Pingle and L. Tesfatsion, *Non-employment benefits and the evolution of worker-employer cooperation: Experiments with real and computational agents*, Economic Report **55** (2001). 52

[118] J. Pollack, A.D. Blair, and M. Land, *Coevolution of a backgammon player*, Proceedings Artificial Life V, MIT Press, 1996, pp. 92–98. 39, 41, 66

[119] C. Ponsati and J. Watson, *Multiple-issue bargaining and axiomatic solutions*, International Journal of Game Theory **26** (1997), no. 4, 501 – 524. 35

[120] H. Raiffa, *The art and science of negotiation*, Harvard University Press, Cambridge, MA, 1982. 28, 29

[121] T. Riechmann, *Genetic algorithm learning and evolutionary games*, Journal of Economic Dynamics and Control **25** (2001), no. 6-7, 1019–1037. 59

[122] J. Romero, *Essays on cooperation and coordination*, Ph.D. thesis, California Institute of Technology, 2010. 15, 56, 121, 124, 125

[123] J.S. Rosenschein and G. Zlotkin, *Rules of encounter: Designing conventions for automated negotiation among computers*, Artificial Intelligence Series, MIT Press, 1994. 34, 35

[124] C.D. Rosin and R.K. Belew, *New methods for competitive coevolution*, Evolutionary Computation **5** (1997), no. 1, 1–29. 38, 42

[125] A.E. Roth, *Independence of irrelevant alternatives, and solutions to Nash's bargaining problem*, Journal of Economic Theory **16** (1977), no. 2, 247–251. 27

[126] ———, *Axiomatic models of bargaining*, Lecture Notes in Economics and Mathematical Systems, no. 170, Springer-Verlag, 1979. 28

[127] A. Rubinstein, *Perfect equilibrium in a bargaining model*, Econometrica **50** (1982), no. 1, 97–109. 32, 33, 77, 81

[128] ———, *A bargaining model with incomplete information about time preferences*, Econometrica **53** (1985), no. 5, 1151–1172. 23

[129] ———, *Modeling bounded rationality*, MIT Press, 1998. 24, 56

[130] J. Rust, J.H. Miller, and R. Palmer, *Characterizing effective trading strategies*, Journal of Economic Dynamics and Control **18** (1994), 61–96. 48

[131] K. Safarzyńska and J.C.J.M. van den Bergh, *Evolutionary models in economics: a survey of methods and building blocks*, Journal of Evolutionary Economics **20** (2010), no. 3, 329–373. 7, 56, 57, 64, 74, 75

[132] T.C. Schelling, *Micromotives and macrobehavior*, WW Norton & Company, 1978. 46, 51

[133] R. Selten, *Re-examination of the perfectness concept for finite points in extensive games*, International Journal of Game Theory **4** (1975), 25–55. 8, 15, 24, 29, 56, 67, 121, 124

[134] ———, *The chain store paradox*, Theory and Decision **9** (1978), no. 2, 127–159. xi, 29, 31

[135] R. Serrano, *Bargaining*, The New Palgrave Dictionary of Economics, McMillan, 2 ed., 2005. 9, 21, 25, 27, 32, 34, 35

[136] H.A. Simon, *A behavioral model of rational choice*, The Quarterly Journal of Economics **69** (1955), no. 1, 99–118. 120

[137] ———, *Rational choice and the structure of the environment.*, Psychological review **63** (1956), no. 2, 129. 55

[138] ———, *Models of man: Social and rational*, John Wiley London, 1957. 6, 24, 54, 55, 120

[139] ———, *Models of bounded rationality*, MIT Press, 1984. 61

[140] K. Sims, *Evolving 3D morphology and behavior by competition*, Artificial Life **1** (1994), no. 4, 353–372. 39, 66

[141] K.O. Stanley and R. Miikkulainen, *The dominance tournament method of monitoring progress in coevolution*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), Morgan Kaufmann, 2002, pp. 242–248. 41

[142] G. Tesauro and J.O. Kephart, *Pricing in agent economies using multi-agent q-learning*, Autonomous Agents and Multi-Agent Systems **5** (2002), no. 3, 289–304. 47, 71

[143] G.J. Tesauro and J.O. Kephart, *Foresight-based pricing algorithms in an economy of software agents*, Proceedings of the First International Conference on Information and Computation Economies, ACM, 1998, pp. 37–44. 47, 163

[144] L. Tesfatsion, *How economists can get alife*, Economic Report 37, Iowa State University Department of Economics, 1995. 3, 51

[145] ———, *Agent-based computational economics: Growing economies from the bottom up*, Artificial life **8** (2002), no. 1, 55–82. 3, 24, 46, 47, 48, 51, 52, 53, 56, 58, 60, 71, 73

[146] W. Thomson, *A class of solutions to bargaining problems*, Journal of Economic Theory **25** (1981), no. 3, 431–441. 27

[147] E. Tsang, *Computational intelligence determines effective rationality*, International Journal of Automation and Computing **5** (2008), no. 1, 63–66. 146

[148] D.D.B. van Bragt, E.H. Gerding, and J.A. La Poutré, *Equilibrium selection in alternating-offers bargaining models - the evolutionary computing approach*, Tech. Report SEN-R0013, CWI, Amsterdam, The Netherlands, December 2000. 96, 120

[149] ———, *Equilibrium selection in alternating-offers bargaining models - the evolutionary computing approach*, The Electronic Journal of Evolutionary Modeling and Economic Dynamics (2002), no. 1027. 47, 54, 69, 74, 81, 82, 88, 96, 120

[150] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*, Princeton University Press, 1944. 22

[151] N.J. Vriend, *Self-organization of markets: An example of a computational approach*, Computational Economics **8** (1995), 205–231. 51

[152] ———, *An illustration of the essential difference between individual and social learning, and its consequences for computational analyses*, Journal of Economic Dynamics and Control **24** (2000), no. 1, 1–19. 4, 48, 59, 63, 72, 73

[153] L.R. Waltman, N.J.P. van Eck, R. Dekker, and U. Kaymak, *Economic modeling using evolutionary algorithms: The effect of a binary encoding of strategies*, Research Paper ERS-2009-028-LIS, Erasmus Research Institute of Management (ERIM), 2010. 52, 62

[154] R.A. Watson and J.B. Pollack, *Coevolutionary dynamics in a minimal substrate*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), Morgan Kaufmann, 2001, pp. 702–709. 39, 41, 66

[155] G. Weisbuch, A. Kirman, and D. Herreiner, *Market organisation and trading relationships*, The Economic Journal **110** (2000), no. 463, 411–436. 51

[156] R.P. Wiegand, *An analysis of cooperative coevolutionary algorithms*, Ph.D. thesis, George Mason University, 2003. 40

[157] O.E. Williamson, *The economics of organization: The transaction cost approach1*, American Journal of Sociology **87** (1981), no. 3, 548–577. 55

[158] Q. Yang, *Game theoretic modeling and analysis: A co-evolutionary, agent-based approach*, Ph.D. thesis, National University of Singapore, 2009. 54

[159] X. Yao, *Evolutionary stability in the n-person iterated prisoner's dilemma*, BioSystems **37** (1996), no. 3, 189–197. 3, 51

[160] D. Zeng and K. Sycara, *Bayesian learning in negotiation*, International Journal of Human Computer Systems **48** (1998), 125–141. 37

Glossary

ad-hoc Not methodical or on an as needed basis, chosen as is from the available tools and techniques which have been developed independently of ACE, specifically when chosen for simulating socio-economic learning. Also called off-the-shelf in the thesis. 3, 324

agents An agent, in this thesis, is a notion that describes a player in a socio-economic game, which uses or is associated with a strategy to play the socio-economic game (e.g. a player playing a bargaining game using a bargaining strategy). 2

bounded rationality Bounded rationality is a concept within the Social Sciences that takes into consideration the cognitive limitations that humans face, in conformance with which they solve problems or make decisions, that lead to outcomes which are unclear or uncertain. The limitations that make one unable to gather, retain, transmit, and/or process information, and the limitations on the amount of time they have to gather, retain, transmit, and/or process information, on route to solving a problem/making a decision, can be seen as limits on rationality, or bounded rationality. For example, a decision maker may not have the resources to find the optimal solution to a problem at hand, which leads to the decision maker making assumptions about the problem in order to arrive at a solution, thus greatly simplifying the problem/possible choices of solutions to the problem. In computational terms, this limitation can translate into an agent being incapable of gathering, retaining, transmitting, and/or processing the information or having limited time to gather, retain, transmit, and/or

process the information that ideally can lead an agent to behaving as if it possessed complete information and unlimited processing capabilities. As such, an agent tries to satisfice rather than optimise its payoff (from a world within which it is situated), i.e. aim for a satisfactory rather than an optimal solution, subject to the information it has been able to gather, retain and/or process, and the time it has to do so. Fully rational behaviour is only possible if decisions can be made such that they lead to outcomes that are clear/certain during the time of processing making such decisions, and that these outcomes do not change whilst processing. In agent based simulations, where agents have limited computational capabilities, such agents are deemed boundedly rational from the outset. 2

co-evolutionary simulations See socio-economic simulations. 10

complete knowledge See Chapter 2, Section 2.2.2. 8

disciplining Providing a methodology for co-evolutionary algorithm analysis and refinements, in turn enabling engineering the algorithm for the purposes of adopting them as socio-economic simulations, is termed as disciplining in the thesis. 8

elements See solution concept. 70

engineering The analysis and refinements of co-evolutionary algorithms for the purposes of modelling socio-economic situations, thus being used as socio-economic simulations, particularly for the case of modelling socio-economic learning within ACE, is termed as engineering in the thesis. 9, 321

equilibrium selection Finding the equilibrium in a game, specifically when it may have multiple equilibria, and thus, one of the many equilibria may have to be chosen as the desired equilibrium, is known as equilibrium selection. 8, 325

evolution strategies An evolution strategy (ES) is an evolutionary search/optimisation algorithm for real valued search, i.e. the candidate solutions to

the search problem are usually represented by a string of real numbers. An objective fitness measure that needs optimising is used in order to evaluate the fitness of candidate solutions. The algorithm usually starts with a population of randomly generated candidate solutions (for example, a string of real numbers), their fitness evaluated using the objective fitness measure. In any generation, individuals are usually selected at random from the pool of candidate solutions and mutated (usually by the addition of a Gaussian distributed noise), to generate a pool of offspring. After evaluating the pool of offspring using the objective fitness measure, the offspring replace the candidate solutions to become the candidate solutions in the next generation, normally via one of the two selection schemes: $(\mu + \lambda)$-ES or (μ, λ)-ES, where μ is the candidate solution pool size and λ, the offspring pool size. In $(\mu + \lambda)$-ES, the fittest μ individuals, amongst both candidate solutions and offspring, become the next generation of candidate solutions. In the (μ, λ)-ES, the fittest μ offspring replace the entire candidate solution pool to be the next generation. This process carries on until a termination condition is satisfied (e.g. a certain number of generations is reached). A detailed discussion of these algorithms is carried out in [17], to which the interested reader is referred to. We are however concerned with the case where these algorithms may be used without an objective fitness measure, i.e. as co-evolutionary algorithms, for the purposes of socio-economic simulations, the particular algorithm covered in this thesis described in Chapter 3, Section 3.5.1.2. 47

evolutionary learning Sub-field of evolutionary computation dealing with using evolutionary algorithms for the purposes of evolving learning machines rather than optimising. 3

genetic algorithms A genetic algorithm (GA) is an evolutionary search/optimisation algorithm inspired by the process of natural evolution. A candidate solution to the search problem is traditionally/canonically represented/encoded as a bit string, but the algorithm is not limited to bit string representations of candidate solutions. An objective fitness measure

that needs optimising is used in order to evaluate the fitness of candidate solutions. The algorithm usually starts with a population of randomly generated candidate solutions (for example, randomly generated bit strings), their fitness evaluated using the objective fitness measure. The fitness values are then used to ascertain the chances (using one of the various selection schemes in the literature [64]) of the candidate solutions being selected for reproduction via variation operators like cross-over and mutation, and this reproduction process leads to the next generation of candidate solutions. The process of evaluation, selection and reproduction usually carries on until a termination condition is satisfied (e.g. a certain number of generations is reached). The interested reader is referred to [64] for more details. There have been many variants of genetic algorithms presented in the literature to date. We are however concerned with the case where these algorithms may be used without an objective fitness measure, i.e. as co-evolutionary algorithms, for the purposes of socio-economic simulations. 3

individual learning A class (also called known as level of learning in ACE) of evolutionary algorithms considered for socio-economic simulations. The defining characteristic of this class is that the evolving population as a whole constitutes a solution (individual being partial solutions) to the problem under question, i.e. the set of individuals (or could be just one individual) within the population represents one agent alone, for example, a learning classifier system. When used as simulations, each agent is thus represented by a set of (or one) individuals (or strategies) which together describe the overall strategy of the agent. For example, if the agent were to represent a socio-economic entity like a *bargaining agent* or a *firm*, the population of individuals (or strategies) belong only to the bargaining agent or the firm and only exchanges information within itself. As such, one bargaining agent or firm could be one learning classifier system. Previous work that has been used as a case study in this thesis (described in Chapter 3, Section 3.5.1.2) also falls into this category, but instead of a learning classifier system, an evolution strategy is used by the bargaining agent. If there are many agents in the simulations, then each agent is represented by

such a set of individuals (or strategies). This is similar to the Michigan approach to evolutionary learning from evolutionary computation literature. See social learning for the other class of algorithms generally used in ACE for socio-economic learning simulations. 3, 326

learning classifier systems A learning classifier system (LCS) is an evolutionary learning algorithm. The system in itself is seen as a learning entity, and has a set of rules that suggest actions given an input stimulus. The rules within the set are matched upon an input stimulus, and the corresponding actions from the matched up rules (there can be multiple matched rules) combined in some manner, giving a resultant action for the LCS to take. Whenever an action is taken, a payoff is received by the LCS from the environment. This payoff is transformed into an accuracy value for the matched up rules, specifically those rules that were more influential in producing the resultant action. Thus, each rule has an accuracy value associated with it, which in fact acts as a fitness value for a genetic algorithm. This genetic algorithm works towards searching for better rules, for example more accurate rules. In essence, the genetic algorithm acts on the set of rules that map stimulus and action, to search for better rules, whilst the rules are credited with a fitness via interacting with the environment the LCS is put in. This is the Michigan approach based learning classifier system. Usually, the Michigan approach to evolutionary learning is what is popularly used to describe learning classifier system. A detailed discussion of these algorithms can be found in [74]. The use of the term 'learning classifier system' in the thesis is to describe a system where each agent may in fact be an LCS of this form, and the environment is described by the agents it interacts with, which then co-adapt, or indeed co-evolve. 47

off-the-shelf Not methodical or on an as needed basis, chosen as is from the available tools and techniques which have been developed independently of ACE, specifically when chosen for simulating socio-economic learning. Also called ad-hoc in the thesis. 2, 320

pareto-efficient The outcome from a game, which would be a payoff pair in

the two play games considered in this thesis, lying on the pareto-efficient frontier, is said to be a pareto-efficient outcome. 25

pareto-efficient frontier When playing a game, the involved players are said to be on the pareto-efficient frontier, if and only if they cannot make mutual gains in their payoffs by individually changing their respective game playing strategies. In a two person game, as considered in this thesis, when players are on the pareto-efficient frontier, if one player individually changes its strategy such that it gets a higher payoff than when using the old strategy, then the other player, who does not change its strategy, will necessarily have its payoff reduced. So, the payoff pairs which have this property are said to lie on the pareto-efficient frontier. 325

perfect rationality See Chapter 2, Section 2.2.2. 8

prisoner's dilemma A two player game, where the players can make one of the two moves, viz. *cooperate* or *defect*. The payoff structure of the game is such that the payoff if the player defects, whilst the other cooperates (this other player is then called the sucker), is higher for the defector, than if both players cooperate. However, if both players defect, it results in a lower payoff for both, as compared to if they cooperated, but still higher than the sucker payoff. They make their moves at the same time, thus not knowing what move the other might make. This leads to the dilemma that the players face, for each is better off defecting (both against a sucker and against a defector), leaving aside the mutual gains from cooperation, which are possible. This is the single round or one shot prisoner's dilemma game. The iterated or repeated form of this game is where the players engage in multiple rounds of this game. The Nash Equilibrium, for the case of both the single and iterated version of the game, is to *always defect*, i.e. defect in every round being played. 23

selected Nash Equilibrium See equilibrium selection. 8

social learning A class (also called known as level of learning in ACE) of evolutionary algorithms considered for socio-economic simulations. The defining

characteristic of this class is that each individual in the evolving population constitutes a solution to the problem under question, i.e. one individual within this population, represents one agent. Genetic algorithms, where one individual in the population may represent one agent, is an example of this class. If an agent were to represent a socio-economic entity like a *bargaining agent* or a *firm*, then the population in the algorithm would be representing a population of bargaining agents or firms, with each individual in the population being the overall strategy of the bargaining agent or firm, which may then exchange information amongst each other, say for example, via the cross-over operator in genetic algorithms. This is similar to the Pittsburg approach to evolutionary learning from evolutionary computation literature. See individual learning for the other class of algorithms generally used in ACE for socio-economic learning simulations. 3, 324

socio-economic learning The process of learning and adaptation by trial and error when a player or agent is made to play or interact within socio-economic environments, for example, interacting, learning and adapting to play bargaining games. In the context of this thesis, the process of co-evolution is seen as modelling this learning and adaptation process, whist the agents play bargaining games. The terms 'socio-economic learning' and 'socio-economic learning and adaptation' are used interchangeably within the thesis and mean the same thing. 2, 326

socio-economic phenomena Socio-economic terms that may get mis-interpreted within co-evolutionary simulations, and as such, the co-evolutionary algorithm may in turn get mis-interpreted based on these terms, are referred to as socio-economic phenomena in the thesis. 3, 326

socio-economic simulations Simulating socio-economic phenomena and socio-economic situations are termed as socio-economic simulations. In the thesis, we are primarily interested in modelling socio-economic learning via co-evolutionary algorithms. So, we primarily mean socio-economic simulations to mean simulating socio-economic learning via co-evolutionary algorithms in the thesis. When we say co-evolutionary simulations, we mean

socio-economic simulations modelling socio-economic learning based on co-evolutionary algorithms. 13, 321, 327

socio-economic situations Situation that may need modelling via socio-economic simulations, for example modelling socio-economic interactions as games and modelling game play using agents that learn and adapt towards various game playing behaviours. Thus, situations which involves socio-economic interaction and socio-economic learning/adaptation, are referred to as socio-economic situations in the thesis. 7, 326

solution concept A game theoretic term that explicitly specifies the nature of the solution one is interested in as being the solution to the game in question. For example, the equilibrium within a game can be seen as a type of solution to the game. For players of the game to be in equilibrium, they have to adhere to game playing strategies with some properties. The specific nature of these strategies that in fact puts the game players in equilibrium with each other, can be seen as the solution concept. In axiomatic game theory, one would state various axioms that should apply to the solution to the game, respecting which might put the players in equilibrium. In strategic game theory, one specifies the general properties one would want in the strategies of the players, and these might put them in equilibrium with each other. Nash Equilibrium is one widely studied solution concept in game theory. In a game like Prisoner's Dilemma, the Nash Equilibrium is for players to defect. The solution concept of Nash Equilibrium for the game thus needs to specify what leads to players defecting. If one puts strategies from within the strategy space of a game through a hypothetical filter such that only the Nash Equilibrium strategy separates out, the filter specifies the solution concept of Nash Equilibrium. The filter searches by way of separating out the desired solution from within the solution space. Similarly, a search algorithm searches through the space of possible solutions to land on to the desired solution. Co-evolutionary algorithms are indeed search algorithms, but due to the nature of these algorithms, in that, the evaluation of strategies at any given time is relative to the opponent strategies discovered thus far, the secondary search problem of the

discovery of opponents comes into play as well. A co-evolutionary solution concept is essentially the specification of the algorithm, such that it expends the necessary effort on the primary and secondary search problems, in its search for the desired solution from the game (e.g. Nash equilibrium solution) when used for finding game playing strategies. Thus, in the thesis, the individual elements (evaluation, interaction, selection, variation and representation) that specify a co-evolutionary algorithm, together specify the co-evolutionary solution concept. 5, 321